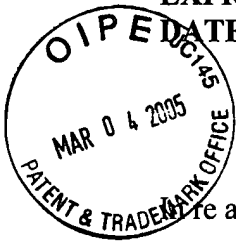


EXPRESS MAIL LABEL NO. EV314044588US

DATE: 04 MARCH 2005



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Re application of inventor(s):

Lukas P.P.P. van Ginneken

Application No. 10/828,547

Filing Date: 19 April 2004

Title: **Unknown**

Group Art Unit: Unknown

Examiner: Unknown

CUSTOMER NO. 36454

MAIL STOP PETITION

Commissioner for Patents

P.O. Box 1450

Alexandria, VA 22313-1450

TRANSMITTAL

Sir:

Enclosed for the referenced file are the following documents:

1. Petition For Suspension Of Examination with Exhibits A-F.
2. Check in the amount of \$400 to cover the Petition Fee.

The Commissioner is hereby authorized to charge any additional fee determined to be due in connection with this communication, or credit any overpayment, to our Deposit Account No. 50-0869 (SYNP 103).

Respectfully submitted,

Dated:

3 March 05

Mark Haynes, Reg. No. 30,846

SYNOPSIS, INC. c/o
HAYNES BEFFEL & WOLFELD LLP
P.O. Box 366
Half Moon Bay, CA 94019
(650) 712-0340 phone
(650) 712-0263 fax

BEST AVAILABLE COPY

EXPRESS MAIL LABEL NO. EV314044588US

MAILED: 04 MARCH 2005

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Re application of inventor(s):

Lukas P.P.P. van Ginneken

Application No. **10/828,547**

Filing Date: **19 April 2004**

Title: **Unknown**

Group Art Unit: Unknown

Examiner: Unknown

CUSTOMER NO. 36454

MAIL STOP PETITION

Commissioner for Patents

P.O. Box 1450

Alexandria, VA 22313-1450

PETITION FOR SUSPENSION OF EXAMINATION

Sir:

Synopsys, Inc. ("Petitioner") petitions under 37 C.F.R. Sections 1.182 and 1.183 to suspend examination for 6 months, or more preferably until resolution of pending litigation in U.S. District Court concerning ownership of U.S. Patent Application No. 10/828,547. In addition, if needed, Petitioner petitions for withdrawal of any outstanding Office Action in U.S. Patent Application No. 10/828,547 for 6 months, or more preferably until resolution of said pending litigation. In the event that there is an outstanding Office Action in U.S. Patent Application No. 10/828,547, Petitioner also petitions to suspend 37 CFR §1.103, which provides that the "Office will not suspend action if a reply by applicant to an Office action is outstanding. Any petition for suspension of action under this paragraph must specify a period of suspension not exceeding six months."

This petition is to be decided by the Office of the Deputy Commissioner for Patent Examination Policy pursuant to MPEP Section 1002.02(b). ("2. Petitions under 37 CFR §1.183 for waiver or suspension of rules not otherwise provided for." and "11. Petitions under 37 CFR §1.182 in matters not otherwise provided for.")

03/09/2005 MMEKONEN 00000022 10828547

01 FC:1462

400.00 DP

STATEMENT OF FACTS

Petitioner believes that U.S. Patent Application No. 10/828,547 was filed by and is being prosecuted by Magma Design Automation, Inc. while in fact Petitioner is the rightful owner. The issue of ownership of U.S. Patent Application No. 10/828,547 is in litigation pending before the U.S. District Court for the Northern District of California as Case No. C04-03923 styled Synopsys, Inc. v. Magma Design Automation, Inc. A copy of the Complaint for Patent Infringement is attached hereto as Exhibit A. A copy of the Answer to Complaint and Counterclaims, filed by Magma Design Automation, Inc. is attached hereto as Exhibit B. Allegations in the Answer to Complaint and Counterclaims raised by Magma Design Automation, Inc. raise the issue of ownership of U.S. Patent Application No. 10/828,547 before the U.S. District Court.

For example, in the Fifth Counterclaim for Relief between paragraphs 124 and 129 of the Answer to Complaint and Counterclaims, the ownership claim of Synopsys to “all of the inventions claimed in the ‘446 Patent and the ‘438 Patent” is stated, and Magma Design Automation, Inc. alleges and seeks a declaratory judgment that Petitioner does not own U.S. Patent Nos. 6,453,446 and 6,725,438, which includes by implication all inventions disclosed therein.

Because U.S. Patent Application No. 10/828,547 claims priority as a continuing application from U.S. Patent Nos. 6,453,446 and 6,725,438, we believe that it necessarily claims an invention disclosed therein. Accordingly, the Fifth Counterclaim for Relief raises the issue of ownership of U.S. Patent Application No. 10/828,547. See also, paragraphs 2 and 3 of the Preliminary Statement of the Answer to Complaint and Counterclaims.

In addition to this petition, Petitioner filed with the United States Patent and Trademark Office on 31 January 2005 revocations of powers of attorney with new powers of attorney and changes of correspondence address in U.S. Patent No. 6,453,446, U.S. Patent No. 6,725,438, and U.S. Patent Application No. 10/828,547 (attached hereto as Exhibits C, D and E, respectively), and recorded assignments from van Ginneken to Synopsys for U.S. Patent No. 6,453,446, U.S. Patent No. 6,725,438, and U.S. Patent Application No. 10/828,547 (attached hereto as Exhibit F), on January 28, 2005. Based on these filings, the dispute concerning ownership is now of record in U.S. Patent Application No. 10/828,547.

SUSPENSION OF EXAMINATION

Suspension of Examination of U.S. Patent Application No. 10/828,547 and Suspension of 37 C.F.R. 1.103 are Appropriate to Allow Pending Federal Litigation to Resolve any Issues of Ownership of U.S. Patent Application No. 10/828,547

A final resolution of Case No. C04-03923 pending before the U.S. District Court for the Northern District of California is expected to resolve the issue of ownership of U.S. Patent Application No. 10/828,547. Suspension of examination of U.S. Patent Application No. 10/828,547 and withdrawal of any pending Office action would therefore preserve the status quo pending determination that the proper party is represented before the United States Patent and Trademark Office during the examination of U.S. Patent Application No. 10/828,547, and that upon conclusion of such examination that a patent resulting from U.S. Patent Application No. 10/828,547 issues to the proper party. Also, suspension of action will prevent the United States Patent and Trademark Office from taking a position inconsistent with the U.S. District Court.

In the event that there is an outstanding Office Action having a statutory deadline, withdrawal of the Office Action would insure that the proper party is represented before the United States Patent and Trademark Office during the examination of U.S. Patent Application No. 10/828,547 upon resolution of the pending litigation. For withdrawal of a pending Office Action, it is also appropriate to suspend the rules of practice, including 37 C.F.R. §1.103, which provides that the "Office will not suspend action if a reply by applicant to an Office action is outstanding. Any petition for suspension of action under this paragraph must specify a period of suspension not exceeding six months."

Accordingly, suspension of examination in U.S. Patent Application No. 10/828,547, withdrawal of any outstanding Office Action in U.S. Patent Application No. 10/828,547, and suspension of 37 C.F.R. §1.103 are appropriate until such litigation is concluded.

CONCLUSION

To preserve the legal and equitable rights in U.S. Patent Application No. 10/828,547, suspension of examination of U.S. Patent Application No. 10/828,547, withdrawal of any outstanding Office Action in U.S. Patent Application No. 10/828,547, and suspension of 37 CFR §1.103 are respectfully requested.

The Petition fee specified in 37 C.F.R. §1.17(f) of \$400 is included herewith.

The Commissioner is hereby authorized to charge any additional fee determined to be due in connection with this communication, or credit any overpayment, to our Deposit Account No. 50-0869 (SYNP 103).

Respectfully submitted,

Dated: 3 Mar 05



Mark Haynes, Reg. No. 30,846

SYNOPSISYS, INC. c/o
HAYNES BEFFEL & WOLFELD LLP
P.O. Box 366
Half Moon Bay, CA 94019
(650) 712-0340 phone
(650) 712-0263 fax

1 Chris Scott Graham (State Bar No. 114498)
2 Michael N. Edelman (State Bar No. 180948)
3 **DECHERT LLP**
4 975 Page Mill Road
5 Palo Alto, California 94304
6 Telephone: (650) 813-4800
7 Facsimile: (650) 813-4848

8 Attorneys for Plaintiff SYNOPSYS

ORIGINAL
FILED

SEP 17 2004

RICHARD W. WIEKING
CLERK U.S. DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
SAN JOSE

9 UNITED STATES DISTRICT COURT
10 NORTHERN DISTRICT OF CALIFORNIA
11 SAN JOSE DIVISION

12 SYNOPSYS, INC., a Delaware corporation

13 Plaintiff,

14 vs.

15 MAGMA DESIGN AUTOMATION, a
16 Delaware corporation,

17 Defendant.

CASE NO. **C04 03923 MEJ**
COMPLAINT FOR PATENT
INFRINGEMENT
ADR
DEMAND FOR JURY TRIAL
E-filing

18
19 Plaintiff SYNOPSYS, INC. ("SYNOPSYS") hereby alleges against Defendant MAGMA
20 DESIGN AUTOMATION ("MAGMA" or "the Defendant") as follows:

21 **JURISDICTION**

22 1. This is an action for patent infringement arising under the patent laws of the United
23 States. This Court has jurisdiction over this action under 28 U.S.C. § 1338(a).

24 **PARTIES**

25 2. SYNOPSYS is a corporation duly organized and existing under the laws of the State
26 of Delaware, with its principal place of business in Mountain View, California.

27 ///

28 ///

9. On April 23, 2002, United States Patent No. 6,378,114 (“the ‘114 Patent”), entitled “Method for the Physical Placement of an Integrated Circuit Adaptive to Netlist Changes,” was issued to SYNOPSYS. van Ginneken is a named inventor on the ‘114 Patent. A true and correct copy of the ‘114 Patent is attached to this complaint as Exhibit B and is incorporated by reference herein.

10. On September 17, 2002, United States Patent No. 6,453,446 ("the '446 Patent"), entitled "Timing Closure Methodology," was issued to MAGMA. The '446 Patent discloses inventions which were made, conceived and developed by van Ginneken at SYNOPSYS. Pursuant to the terms of the Agreement, SYNOPSYS holds legal and equitable title to the inventions in the '446 Patent. A true and correct copy of the '446 Patent is attached to this complaint as Exhibit C and is incorporated by reference herein.

11. On April 20, 2004, United States Patent No. 6,725,438 ("the '438 Patent"), entitled "Timing Closure Methodology," was issued to MAGMA. The '438 Patent contains inventions which were made, conceived and developed by van Ginneken at SYNOPSYS. Pursuant to the terms of the Agreement, SYNOPSYS holds legal and equitable title to the inventions in the '438 Patent. A true and correct copy of the '438 Patent is attached to this complaint as Exhibit D and is incorporated by reference herein.

12. Since the issuance of the '114 Patent, '446 Patent, and '438 Patent (collectively referred to hereinafter as the "SYNOPSISYS PATENTS"), MAGMA has engaged in a wide range of activities to infringe those patents. MAGMA has been involved in making, using, selling, distributing, advertising, marketing and creating source code for products that infringe the SYNOPSISYS PATENTS.

FIRST CAUSE OF ACTION
(PATENT INFRINGEMENT)

13. SYNOPSISYS is the owner of the SYNOPSISYS PATENTS because, among other reasons, the inventions disclosed in the patents were previously assigned to SYNOPSISYS by van Ginneken pursuant to the terms of the Agreement.

111

1 14. MAGMA has been and still is infringing the SYNOPSIS PATENTS in violation of
2 the federal patent laws by making, using, selling, distributing, advertising, marketing and creating
3 source code for products which infringe the SYNOPSIS PATENTS. MAGMA will continue to so
4 infringe unless enjoined by this Court.

5 15. MAGMA has actively induced infringement of, or contributed to the infringement of,
6 the SYNOPSIS PATENTS under the federal patent laws by, among other things, making infringing
7 products and creating source code for infringing products and then selling, distributing, advertising
8 and marketing those infringing products to others, and will continue to do so unless enjoined by this
9 Court.

10 16. MAGMA's infringement of the SYNOPSIS PATENTS in violation of the federal
11 patent laws has been willful and deliberate, and has caused injury to SYNOPSIS.

12 17. MAGMA's infringement in violation of the federal patent laws will continue to injure
13 SYNOPSIS unless enjoined by this Court.

14 WHEREFORE, SYNOPSIS prays for judgment against the Defendant, and requests that this
15 Court impose the following remedies under the federal patent laws:

16 A. Preliminarily and permanently enjoin the Defendant from continued infringement of
17 the SYNOPSIS PATENTS, pursuant to 35 U.S.C. § 283;

18 B. Order the Defendant to account to SYNOPSIS for damages sustained by
19 SYNOPSIS as a result of the Defendant's infringement of the SYNOPSIS PATENTS, with
20 interest, pursuant to 35 U.S.C. § 284;

21 C. Order the Defendant to pay SYNOPSIS a reasonable royalty to compensate for the
22 Defendant's infringement, pursuant to 35 U.S.C. § 284;

23 D. Treble the damages resulting from the Defendant's willful and deliberate
24 infringement, pursuant to 35 U.S.C. § 284;

25 E. Award SYNOPSIS its costs, expenses and reasonable attorneys' fees incurred in
26 bringing and prosecuting this action, pursuant to 35 U.S.C. § 285;

27 ///


28 ///

1 F. Impose a constructive trust for the benefit of SYNOPSYS over any profits, revenues,
2 or other benefits obtained by the Defendant as a result of its infringement of the SYNOPSYS
3 PATENTS; and

4 G. Award SYNOPSYS such further relief that the Court may deem just and proper
5 arising from the Defendant's infringement of the SYNOPSYS PATENTS under the federal patent
6 laws.

7 Dated: September 17, 2004

DECHERT LLP

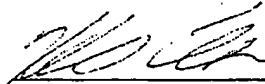
8 
9 Chris Scott Graham
10 Michael Edelman
11 Attorneys for Plaintiff SYNOPSYS
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

DEMAND FOR JURY TRIAL

SYNOPSYS hereby demands trial by jury of all issues.

Dated: September 17, 2004

DECHERT LLP



Chris Scott Graham

Michael Edelman

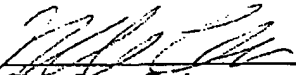
Attorneys for Plaintiff SYNOPSYS

CERTIFICATION OF INTERESTED ENTITIES OR PERSONS

Pursuant to Civil L.R. 3-16, the undersigned certifies that as of this date, other than the named parties, there is no such interest to report.

Dated: September 17, 2004

DECHERT LLP


Chris Scott Graham
Michael Edelman
Attorneys for Plaintiff SYNOPSISYS

PROPRIETARY INFORMATION
AND
INVENTIONS AGREEMENT

The following confirms an agreement between me and Synopsys, Inc. (the "Company"), which is a material part of the consideration for my employment by the Company.

1. I recognize that the Company is engaged in a continuous program of research, development and production respecting its business, present and future, including fields generally related to its business and that the Company possesses and continues to possess information that has been created, discovered, developed or otherwise become known to the Company (including, without limitation, information created, discovered or developed by, or made known to, me during the period of or arising out of my employment by the Company) and/or in which property rights have been assigned, licensed or otherwise conveyed to the Company, which information has commercial value in the business in which the Company is engaged. All of the aforementioned information is hereinafter called "Proprietary Information." By way of illustration, but not limitation, Proprietary Information includes trade secrets, processes, data and know-how, computer software, improvements, inventions, works of authorship, techniques, marketing plans, strategies, forecasts and copyrightable material and customer lists.

2. I understand that my employment creates a relationship of confidence and trust between me and the Company with respect to any information:

(i) applicable to the business of the Company; or

(ii) applicable to the business of any client or customer of the Company, which may be known to me by the Company or by any client or customer of the Company, or learned by me during the period of my employment.

3. In consideration of my employment by the Company and the compensation received by me from the Company from time to time, I hereby agree as follows:

A. All Proprietary Information shall be the sole property of the Company and its assigns, and the Company and its assigns shall be sole owner of all patents, copyrights and other rights in connection therewith. I hereby assign to the Company any rights I may have or acquire in such Proprietary Information. At all times, both during my employment by the Company and after its termination, I will keep in confidence and trust all Proprietary Information, and I will not use or disclose any Proprietary Information or anything relating to it without the written consent of the Company, except as may be necessary in the ordinary course of performing my duties to the Company.

B. All documents, records, apparatus, equipment and other physical property, whether or not pertaining to Proprietary Information, furnished to me by the Company or produced by me or others in connection with my employment shall be and remain the sole property of the Company and shall be returned to the Company immediately as and when requested by the Company. Even if the Company does not so request, I shall return and deliver all such property upon termination of my employment by me or the Company for any reason and I will not take with me any such property or any reproduction of such property upon such termination.

C. I will promptly disclose to the Company, or any persons designated by it, all improvements, inventions, works of authorship, processes, techniques, know-how, formulae,

data, ideas and other information (including, without limitation, my algorithms or software), whether or not patentable, made or conceived or reduced to practice or learned by me, either alone or jointly with others, during the term of my employment (all said improvements, inventions, works of authorship, processes, techniques, know-how, formulae, data, ideas and other information shall be hereinafter collectively called "Inventions").

D. I agree that all Inventions which I make, conceive, reduce to practice or develop (in whole or in part, either alone or jointly with others) during my employment shall be the sole property of the Company to the maximum extent permitted by Section 2870 of the California Labor Code (hereinafter called "Section 2870"), a copy of which is attached hereto as Exhibit B, and to the extent permitted by law, shall be "works made for hire." The Company shall be the sole owner of all patents, copyrights, trade secret rights, rights with respect to other intellectual property or other rights in connection therewith (including, without limitation, such rights in algorithms or software). I hereby assign to the Company any rights I may have or acquire in such Inventions. I agree to perform, during and after my employment, all acts deemed necessary or desirable by the Company to permit and assist it, at the Company's expense, in obtaining, maintaining and enforcing patents, copyrights, trade secret rights, rights with respect to such Inventions and/or other Inventions I have or may at any time assign to the Company in any and all countries. Such acts may include, but are not limited to, execution of documents and assistance or cooperation in legal proceedings. I hereby irrevocably designate and appoint the Company and its duly authorized officers and agents as my agents and attorneys-in-fact to act for and on my behalf and instead of me, to execute and file any applications or related filings and to do all other lawfully permitted acts to further the prosecution, maintenance and enforcement, issuance of patents, copyrights, trade secret rights, rights with respect to mask works or other rights thereon with the same legal force and effect as if executed by me.

E. As a matter of record, I attach hereto a complete list of all inventions or improvements relevant to the subject matter of my employment by the Company which have been made by me or jointly with others prior to my employment with the Company that I desire to remove from the operation of this Agreement, and I covenant that such list is complete. If no such list is attached to this Agreement, I represent that I have no such inventions and improvements at the time of signing this Agreement.

F. During the term of my employment and for one (1) year thereafter, I will not encourage or solicit any employee of the Company to leave the Company for any reason or devote less than all of any such employee's efforts to the affairs of the Company, provided that the foregoing shall not affect any responsibility I may have as an employee if the Company with respect to the bona fide hiring and firing of Company personnel.

G. I represent that my performance of all the terms of this Agreement will not breach any agreement to keep in confidence proprietary information acquired by me in confidence or in trust prior to my employment by the Company. I have not entered into, and I agree I will not enter into, any agreement, either written or oral, in conflict herewith.

H. I represent that execution of this Agreement, my employment with the Company and my performance of my proposed duties to the Company in the development of its business will not violate any obligations I may have to my former employer.

I. This Agreement does not require assignment of an invention which an employee cannot be obligated to assign under Section 2870. However, I will disclose any Inventions as required by Section 3(c) hereof regardless of whether I believe the Invention is protected by Section 2870, in order to permit the Company to engage in a review process to determine such issues as may arise. Such disclosure shall be received in confidence by the Company.

4. This Agreement shall be effective as of the first day of my employment by the Company: June 26 1995 (date).

5. This Agreement shall be binding upon me, my heirs, executors, assigns and administrators and shall inure to the benefit of the Company, its successors and assigns.

Dated: 5/17/95


Signature

Lukas van Ginneken
(Print or Type Name)

Accepted and agreed to:

Synopsys, Inc.

By  _____

EXHIBIT A

TO

SYNOPSYS, INC. PROPRIETARY INFORMATION
AND INVENTIONS AGREEMENT

1. The following is a complete list of all inventions or improvements relevant to the subject matter of my employment by Synopsys, Inc. (the "Company") that have been made or conceived or first reduced to practice by me alone or jointly with others prior to my employment by the Company that I desire to remove from the operation of the Company's Proprietary Information and Inventions Agreement.

☐ No inventions or improvements

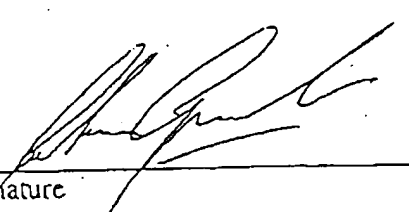
☐ See below

☒ Additional sheets attached

2. I propose to bring to my employment the following materials and documents of a former employer:

☒ No materials or documents

☐ See below


Signature

Lukas van Gyncken
Print or Type Name

EXHIBIT B

TO

SYNOPSYS, INC. PROPRIETARY INFORMATION
AND INVENTIONS AGREEMENT

Section 2870. Application of provision providing that employee shall assign or offer to assign rights in inventions to employer.

(a) Any provision in an employment agreement which provides that an employee shall assign, or offer to assign, any of his or her rights in an invention to her or her employer shall not apply to an invention that the employee entirely on his or her own time without using the employer's equipment, supplies, facilities or trade secret information except for those inventions that either:

(1) Relate at the time of conception or reduction to practice of the invention to the employer's business, or actual or demonstrably anticipated research or development by the employer;

(2) result from any work performed by the employee for the employer.

(b) To the extent a provision in an employment agreement purports to require an employee to assign an invention otherwise excluded from being required to be assigned under subdivision (a), the provision is against the public policy of this state and is unenforceable.

The following is a complete list of all inventions or improvements relevant to the subject matter of my employment by Synopsys, Inc. (the Company) that have been made or conceived or first reduced to practice by me alone or jointly with others prior to my employment by the Company that I desire to remove from the operation of the Company's Proprietary Information and Inventions Agreement.

[0] "Efficient orthonormality testing for synthesis with pass transistor selectors" by M. R. C. M. Berkelaar and -, accepted at the International Workshop on Logic Synthesis, June 1995.

[1] "Timing Verification and Optimization for the PowerPC Processor Family", by R.E. Mains, T. A. Mosher, - and R.F. Damiano, in: Proc. Int. Conf. on Computer Design, pp.390-393, Boston, Oct. 10-12, 1994.

[2] "In the driver's seat of BooleDozer" by D. Brand and R.F. Damiano, -, A. D. Drumm, in: Proc. Int. Conf. on Computer Design, pp. 518-521 Boston, Oct. 10-12, 1994.

[3] "Grammar-based optimization of synthesis scenarios" by A. Kuehlmann and -, in: Proc. Int. Conf. on Computer Design, pp. 20-25 Boston, Oct. 10-12, 1994.

[4] "Tuning of logic synthesis scenarios" by - and A. Kuehlmann, Workshop notes of the Int. Workshop on logic synthesis, paper P7c, Tahoe City, May 23-26, 1993.

[5] "Fanin ordering in multi-slot timing" by -, Proc. Int. Conf. on Computer Design, pp. 44-47, Cambridge, Oct. 11-14, 1992.

[6] "The complexity of adaptive annealing" by R. H. J. M. Otten and -, Proc. Int. Conf. on Computer Design, pp. 404-407, Cambridge, Sept. 17-19, 1990.

[7] "Buffer placement in distributed RC-tree networks for minimal Elmore delay" by -, Proc. Int. Symp. on Circuits and Systems, pp. 865-868, New Orleans, May 2-5, 1990.

[8] "Optimal slicing of plane point placements" by - and R. H. J. M. Otten, Proc. European Design Automation Conf. pp. 322-236, Glasgow, March 12-15, 1990.

[9] The annealing algorithm by R. H. J. M. Otten and -, ISBN 07923-9022-9, Boston:Kluwer, 1989.

[10] The predictor-adaptor paradigm - automation of custom layout by flexible design by -, Ph.D. thesis, ISBN 90-9002703-3, Eindhoven, 1989.

[11] "Doubly folded transistor matrix layout" by - and J. T. J. van Eijndhoven, A. H. C. M. Brouwers, Digest Int. Conf. on Computer Aided Design, Santa Clara, Nov. 7-10, 1988.

[12] "Stop criteria in simulated annealing" R. H. J. M. Otten and -, Proc.

Int. Conf. on Computer Design, pp.549-552, Port Chester, Oct. 3-5,
1988.

[13] "An inner loop criterion for simulated annealing" by - and R.H.J.M.
Otten, Physics letters A, 130:429-435, 1988.

[14] "Soft Macro Cell generation by two dimensional folding" by - and J.
T. J. van Eijndhoven, P. R. M. van Teeffelen, T. J. Deckers, Proc. Int.
Symp. on Circuits and Systems, pp. 727-730, Espoo, June 1988.

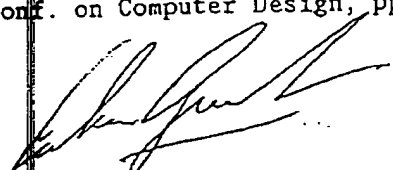
[15] "Gridless routing of general floor plans" by - and J. A. G. Jess, Digest
Int. Conf. on Computer Aided Design, pp. 30-33, Santa Clara Nov. 9-
12, 1987.

[16] "Wire planning for stackable designs", by R. K. Brayton, C. L. Chen,
J. A. G. Jess, R. H. J. M. Otten and -, Proc. Int. Symp. on VLSI tech-
nology, pp.269-273, Taipeh, May 13-15, 1987.

[17] "Global wiring for custom layout design" by - and R. H. J. M. Otten,
Proc. Int. Symp. on Circuits and Systems. pp.207-208, Kyoto, June 5-
7, 1985.

[18] "Floor plan design using simulated annealing" by R. H. J. M. Otten
and -, Digest Int. Conf. on Computer Aided Design, pp. 96-98, Santa
Clara, Nov. 1984.

[19] "Stepwise layout refinement" by - and R. H. J. M. Otten, Proc. Int.
Conf. on Computer Design, pp. 30-36, Port Chester, Oct.8-11, 1984.



Lukas van Ginneken

1 GEORGE A. RILEY (S.B. #118304)
2 CHRISTOPHER D. CATALANO (S.B. #208606)
3 RYAN K. YAGURA (S.B. #197619)
4 LUANN L. SIMMONS (S.B. #203526)
5 O'MELVENY & MYERS LLP
6 Embarcadero Center West
7 275 Battery Street
8 San Francisco, California 94111-3305
9 Telephone: (415) 984-8700
10 Facsimile: (415) 984-8701

11 Attorneys for Defendant and Counterclaimant
12 MAGMA DESIGN AUTOMATION, INC.

13 UNITED STATES DISTRICT COURT
14 NORTHERN DISTRICT OF CALIFORNIA
15 SAN FRANCISCO DIVISION

16 SYNOPSISYS, INC., a Delaware
17 Corporation,

18 Plaintiff and
19 Counter-Defendant,

20 v.

21 MAGMA DESIGN AUTOMATION,
22 INC., a Delaware Corporation

23 Defendant and
24 Counterclaimant.

25 AND RELATED COUNTERCLAIMS.

Case No. C04-03923 MMC

**DEFENDANT MAGMA DESIGN
AUTOMATION, INC.'S ANSWER TO
COMPLAINT AND
COUNTERCLAIMS**

DEMAND FOR JURY TRIAL

26 Defendant Magma Design Automation, Inc. ("Magma"), by and through its
27 attorneys, alleges on knowledge as to its own conduct and on information and belief as to
28 all other matters, as follows:

PRELIMINARY STATEMENT

1. Synopsys, Inc. ("Synopsys") has sued Magma for allegedly
infringing three patents relating to electronic design automation ("EDA") technology.
Only one of the patents, U.S. Patent No. 6,378,114 ("the '114 Patent"), is actually

MAGMA'S ANSWER TO COMPLAINT
AND COUNTERCLAIMS
Case No. C04-03923 MMC

1 assigned to Synopsys. Magma's innovative products, however, are fundamentally
2 different from the technology claimed in the '114 Patent. Thus, Magma does not infringe
3 the '114 Patent.

4 2. Magma – not Synopsys – is the assignee and the sole and exclusive
5 owner of the other two patents asserted here, U.S. Patents Nos. 6,453,446 ("the '446
6 Patent") and 6,725,438 ("the '438 Patent") (together, "the Magma Patents"). In a
7 complaint devoid of any facts, Synopsys claims ownership of the Magma Patents based
8 entirely on the conclusory allegation that Dr. Lukas van Ginneken created the inventions
9 disclosed in the Magma Patents while he was employed at Synopsys.

10 3. The truth is far different. As detailed in the factual allegations below,
11 Dr. van Ginneken developed the inventions disclosed in the Magma Patents after he co-
12 founded Magma in 1997. Drawing on his superlative academic background and extensive
13 industry experience, Dr. van Ginneken created these inventions without using proprietary
14 or confidential information from Synopsys. Thus, Synopsys' claim of ownership to the
15 Magma Patents is groundless.

16 4. Synopsys' interest in Magma's technology is a recent transformation.
17 From its beginnings, Magma has described its novel technology to Synopsys during
18 various meetings. Although Synopsys professed interest in Magma's talented engineering
19 team, Synopsys repeatedly, both in public and in private, denigrated Magma's technology.
20 Not once during these discussions did Synopsys ever assert any ownership interest in that
21 technology.

22 5. During the past two years, however, Magma has become a
23 competitive threat to Synopsys. Prompted by this change in the competitive landscape,
24 Synopsys, which holds a dominant position in many EDA markets, has launched a
25 campaign to discredit Magma in the eyes of Magma's customers and investors. As part of
26 this campaign, Synopsys filed this baseless action and has tried to use it to disrupt
27 Magma's relationships with its customers. As the facts demonstrate, however, Magma
28

1 does not infringe the '114 Patent, and Magma holds all rights, title, and interest in and to
2 the Magma Patents.

3 6. Synopsys' claims will fail for an additional reason: As a matter of
4 law, Magma may not be held liable for alleged infringement of the '114 Patent or the
5 Magma Patents.

6 7. Dr. van Ginneken's work at Synopsys that led to the development of
7 the inventions claimed in the '114 Patent was conducted as part of a project between IBM
8 and Synopsys governed by a joint development agreement. Under the terms of the
9 agreement, IBM and Synopsys share ownership of inventions resulting from the project.
10 Synopsys, however, failed to acknowledge IBM during prosecution of the '114 Patent.
11 By operation of law and pursuant to the joint development agreement, IBM is a co-owner
12 of the '114 Patent. Because all co-owners of a patent must be joined as plaintiffs in an
13 infringement action, Synopsys' failure to name IBM as a plaintiff in this suit is fatal to
14 Synopsys' claim for infringement of the '114 Patent.

15 8. The same result would apply to the Magma Patents if – contrary to
16 the facts – Synopsys could somehow establish that the inventions in the Magma Patents
17 were conceived by Dr. van Ginneken while he was at Synopsys. All the work that Dr. van
18 Ginneken did at Synopsys in the areas of logic synthesis and physical design was part of
19 the joint project with IBM. Thus, even if Synopsys were to prevail on its ownership
20 claims, IBM would be a co-owner of the Magma Patents by operation of law and pursuant
21 to the IBM-Synopsys joint development agreement. In that case, Synopsys' failure to
22 name IBM as a plaintiff in this suit is fatal to Synopsys' claim for infringement of the
23 Magma Patents.

24 9. Synopsys' infringement claims fail as a matter of law for another
25 reason. On March 24, 2004, Magma and IBM entered into a patent license agreement.
26 Under this license agreement, Magma is broadly licensed to all patents owned by IBM
27 that were filed before a specified date. The '114 and the Magma Patents all were filed
28 before that date. Magma is therefore licensed under the '114 Patent. Furthermore, if

1 Synopsys could somehow show that the inventions in the Magma Patents were conceived
2 by Dr. van Ginneken at Synopsys, Magma is licensed under those patents as well.

3 10. In its counterclaims, Magma affirms its exclusive ownership of the
4 Magma Patents. Magma further seeks declaratory judgments that Magma cannot infringe
5 any of the three patents asserted by Synopsys.

6 11. Magma also seeks relief under section 17200 *et seq.* of the California
7 Business and Professions Code to enjoin Synopsys' campaign to spread false and
8 misleading statements about Magma and its products. Customers should have a full and
9 fair opportunity, free from misrepresentations and manipulation, to choose the best
10 products based on performance.

11 **MAGMA'S ANSWER TO SYNOPSIS' COMPLAINT**

12 Magma, by and through its attorneys, answers the Complaint for Patent
13 Infringement (the "Complaint") of Synopsys as follows:

14 12. Magma denies each and every allegation contained in paragraph 1,
15 except that Magma admits that this Court has subject matter jurisdiction over this action.

16 13. Magma alleges it is without information or knowledge sufficient to
17 form a belief as to the truth of the allegations in paragraph 2, and on that basis denies each
18 and every allegation contained therein.

19 14. Magma admits the allegations contained in paragraph 3.

20 15. Magma denies each and every allegation contained in paragraph 4,
21 except that Magma admits that venue is proper in the Northern District of California.

22 16. Magma admits the allegations in the first sentence of paragraph 5.
23 Magma alleges that it is without information or knowledge sufficient to form a belief as to
24 the truth of the allegations in the second sentence of paragraph 5, and on that basis denies
25 each and every allegation contained therein.

26 17. Magma admits that Lukas van Ginneken signed a Proprietary
27 Information and Inventions Agreement with Synopsys. Magma alleges that it is without
28 information or knowledge sufficient to form a belief as to the truth of Synopsys'

1 allegation that the document attached as Exhibit A to the Complaint is a true and correct
2 copy of a Proprietary Information and Inventions Agreement between Lukas van
3 Ginneken and Synopsys, and on that basis denies that allegation. Magma denies all the
4 remaining allegations contained in paragraph 6.

5 18. Magma denies each and every allegation contained in paragraph 7.

6 19. Magma admits the allegations contained in the first sentence of
7 paragraph 8. Magma denies each and every allegation contained in the second sentence of
8 paragraph 8.

9 20. Magma admits that Synopsys is listed as the assignee on the face of
10 U.S. Patent No. 6,378,114 and that Lukas van Ginneken is a named inventor of this
11 patent. Magma admits that a true and correct copy of the '114 Patent as issued by the
12 U.S. Patent and Trademark Office is attached to the Complaint as Exhibit B. Magma
13 alleges it is without information or knowledge sufficient to form a belief as to the truth of
14 the remaining allegations contained in paragraph 9, and on that basis denies each and
15 every allegation contained therein.

16 21. Magma denies each and every allegation contained in paragraph 10,
17 except that Magma admits that U.S. Patent No. 6,453,446 was issued to Magma on
18 September 17, 2002.

19 22. Magma denies each and every allegation contained in paragraph 11,
20 except that Magma admits that U.S. Patent No. 6,725,438 was issued to Magma on April
21 20, 2004, and that a true and correct copy of the '438 Patent as issued by the U.S. Patent
22 and Trademark Office is attached to the Complaint as Exhibit D.

23 23. Magma denies each and every allegation contained in paragraph 12.

24 **FIRST CAUSE OF ACTION**

25 **(PATENT INFRINGEMENT)**

26 24. Magma denies each and every allegation contained in paragraph 13.

27 25. Magma denies each and every allegation contained in paragraph 14.

28 26. Magma denies each and every allegation contained in paragraph 15.

1 27. Magma denies each and every allegation contained in paragraph 16.
2 28. Magma denies each and every allegation contained in paragraph 17.
3 29. In response to the prayer for relief, Magma denies each and every
4 allegation in the prayer and, further, Magma specifically denies that Synopsys is entitled
5 to any of the relief requested in the Complaint or any relief whatsoever, specifically
6 denies that Synopsys is entitled to preliminary or permanent injunctive relief, specifically
7 denies that Synopsys has been damaged by the acts of Magma in any amount whatsoever,
8 specifically denies that Synopsys is entitled to an accounting for its alleged damages,
9 specifically denies that Synopsys is entitled to a reasonable royalty, specifically denies
10 that Synopsys is entitled to any award of treble, punitive, or exemplary damages,
11 specifically denies that Synopsys is entitled to its costs, expenses or reasonable attorneys'
12 fees, specifically denies that Synopsys is entitled to any award of interest, and specifically
13 denies that the Court should impose a constructive trust for Synopsys' benefit.

14

15

AFFIRMATIVE DEFENSES

16

AFFIRMATIVE DEFENSES APPLICABLE TO THE '114 PATENT

17

FIRST AFFIRMATIVE DEFENSE

18

19

30. Magma does not infringe, or contribute to or induce the infringement
of, the '114 Patent.

20

SECOND AFFIRMATIVE DEFENSE

21

22

31. Synopsys lacks standing to assert the '114 Patent for failure to join
all joint owners.

23

THIRD AFFIRMATIVE DEFENSE

24

25

32. Magma cannot be liable for infringing the '114 Patent because
Magma is licensed under the '114 Patent.

26

FOURTH AFFIRMATIVE DEFENSE

27

28

33. The '114 Patent is invalid because it fails to satisfy the conditions for
patentability specified in Title 35 of the United States Code.

1 **AFFIRMATIVE DEFENSES APPLICABLE TO THE MAGMA PATENTS**

2 **FIFTH AFFIRMATIVE DEFENSE**

3 34. Synopsis lacks standing to assert the '446 Patent because Magma
4 owns all right, title, and interest in and to the '446 Patent.

5 **SIXTH AFFIRMATIVE DEFENSE**

6 35. In the alternative, if Magma does not exclusively own the '446
7 Patent, Synopsis lacks standing to assert the '446 Patent for failure to join all joint
8 owners.

9 **SEVENTH AFFIRMATIVE DEFENSE**

10 36. In the alternative, if Magma does not own the '446 Patent, Magma
11 does not infringe, or contribute to or induce the infringement of, the '446 Patent.

12 **EIGHTH AFFIRMATIVE DEFENSE**

13 37. In the alternative, if Magma does not own the '446 Patent, Magma
14 cannot be liable for infringement of the '446 Patent because Magma is licensed under the
15 '446 Patent.

16 **NINTH AFFIRMATIVE DEFENSE**

17 38. In the alternative, if Magma does not own the '446 Patent, the '446
18 Patent is invalid because it fails to satisfy the conditions for patentability specified in Title
19 35 of the United States Code.

20 **TENTH AFFIRMATIVE DEFENSE**

21 39. Synopsis lacks standing to assert the '438 Patent because Magma
22 owns all right, title, and interest in and to the '438 Patent.

23 **ELEVENTH AFFIRMATIVE DEFENSE**

24 40. In the alternative, if Magma does not exclusively own the '438
25 Patent, Synopsis lacks standing to assert the '438 Patent for failure to join all joint
26 owners.

1 **TWELFTH AFFIRMATIVE DEFENSE**

2 41. In the alternative, if Magma does not own the '438 Patent, Magma
3 does not infringe, or contribute to or induce the infringement of, the '438 Patent.

4 **THIRTEENTH AFFIRMATIVE DEFENSE**

5 42. In the alternative, if Magma does not own the '438 Patent, Magma
6 cannot be liable for infringement of the '438 Patent because Magma is licensed under the
7 '438 Patent.

8 **FOURTEENTH AFFIRMATIVE DEFENSE**

9 43. In the alternative, if Magma does not own the '438 Patent, the '438
10 Patent is invalid because it fails to satisfy the conditions for patentability specified in Title
11 35 of the United States Code.

12 **FIFTEENTH AFFIRMATIVE DEFENSE**

13 44. In the alternative, Synopsys' claim to ownership of the Magma
14 Patents is barred under 35 U.S.C. § 261.

15 **AFFIRMATIVE DEFENSES APPLICABLE TO ALL PATENTS-IN-SUIT**

16 **SIXTEENTH AFFIRMATIVE DEFENSE**

17 45. Synopsys' claims are barred by the doctrine of laches.

18 **SEVENTEENTH AFFIRMATIVE DEFENSE**

19 46. Synopsys' claims are barred by the doctrine of waiver.

20 **EIGHTEENTH AFFIRMATIVE DEFENSE**

21 47. Synopsys' claims are barred by the doctrine of estoppel.

22 **NINETEENTH AFFIRMATIVE DEFENSE**

23 48. Synopsys' claims are barred by applicable statutes of limitations.

24
25 **MAGMA'S COUNTERCLAIMS AGAINST SYNOPSYS**

26 For its counterclaims against Synopsys, defendant and counterclaimant
27 Magma alleges on knowledge as to its own conduct and on information and belief as to all
28 other matters, as follows:

1
2
3
4
5
6
7
8
9
10

JURISDICTION

49. This action arises under the patent laws of the United States, 35 U.S.C. § 100, *et seq.* Subject matter jurisdiction is therefore proper in this Court pursuant to 28 U.S.C. §§ 1331, 1338 and 1367(a) and pursuant to the Federal Declaratory Judgments Act, 28 U.S.C. §§ 2201-02. This Court has supplemental jurisdiction over Defendant's counterclaims arising under the state law pursuant to 28 U.S.C. § 1367(a) because these claims are so related to the parties' claims and counterclaims under federal law that they form part of the same case and/or controversy and derive from a common nucleus of operative fact.

11
12
13
14
15
16
17

PARTIES

50. Magma is a corporation organized and existing under the laws of the State of Delaware and has its principal place of business in Santa Clara, California. Magma provides EDA software products and related services.

51. Synopsys is a corporation organized and existing under the laws of the State of Delaware and has its principal place of business in Mountain View, California. Synopsys provides EDA software products and related services.

18
19
20
21
22
23
24

VENUE

52. Synopsys transacts business in this judicial district, including the sale and offering for sale of its products, and Synopsys has sufficient contacts with this judicial district to subject itself to the jurisdiction of this Court. Moreover, by bringing its complaint against Magma in this Court, Synopsys consented to the Court's jurisdiction. Personal jurisdiction and venue are therefore proper in this Court pursuant to 28 U.S.C. §§ 1391 and 1400(b).

25
26
27
28

FACTS RELEVANT TO MAGMA'S COUNTERCLAIMS

ELECTRONIC DESIGN AUTOMATION

53. EDA companies develop computer programs that are used to design, manufacture, and test integrated circuits ("ICs" or "chips"). These programs are crucial to the growth of the semiconductor industry.

1 54. It would be impossible to design modern ICs without advanced EDA
2 software. Feature density, speed, efficiency, and functional capacity of ICs continue to
3 increase at a dramatic rate. Current generation ICs hold tens of millions of transistors and
4 have feature widths of 130 nanometers (billionths of a meter) and below. ICs such as
5 microprocessors can execute hundreds of millions of instructions every second.

6 55. IC designers use EDA software to translate high level descriptions of
7 an IC into the basic components that will be fabricated on the chip. IC designers also use
8 EDA software to create a detailed physical layout of the chip, precisely locating within the
9 chip's boundary each of the components and the wires that interconnect them. This
10 physical layout is used to create the mask that is used to fabricate the circuits that make up
11 a chip. Because a modern IC comprises millions of basic components and wires, the
12 design process requires extraordinarily powerful EDA software to ensure that the final IC
13 layout meets the designer's specifications.

14 56. The process of translating an IC design from a high level description
15 to a physical layout is not merely a conversion of the design from one representation to
16 another. At various stages, EDA software optimizes the design in order, for example, to
17 improve the chip's performance, reduce the chip's size, or decrease the chip's power
18 consumption. These optimizations are critical to the overall design process.

19 57. Developers of EDA software – such as Synopsys and Magma –
20 compete with each other based on the quality of the optimizations offered by their
21 products. In competitive “benchmarks” of EDA products, customers generally choose
22 the software that produces the best results while requiring less time and fewer engineers.

23 **THE INTEGRATED CIRCUIT DESIGN PROCESS**

24 58. The design process for an IC may be divided into two basic parts:
25 “front-end” design tasks, referred to as “logic synthesis,” and “back-end” design tasks,
26 referred to as “physical design.”

27 59. *Logic Synthesis.* Logic synthesis refers to the translation of high
28 level descriptions of the functions that the IC must perform into basic logical operations.

1 The high level descriptions, referred to as register transfer level ("RTL") specifications,
2 can be written directly by a design engineer or can be generated by a software program.
3 In the logic synthesis phase, EDA software tools convert the RTL specifications into an
4 interconnected set of logic gates. (A logic gate performs a simple logical function, such
5 as comparing two signals and producing a result.) The tools produce a data file known as
6 a "netlist" that describes the logic gates and their interconnections. The design must be
7 "mapped" to an IC manufacturer's "cell library," which is made up of pre-designed
8 groups of transistors that perform the functions of the gates. The logic synthesis stage
9 typically includes timing analysis to determine approximately how fast the IC will run.

10 60. *Physical Design.* Physical design uses the mapped netlist produced
11 by the logic synthesis phase to determine the actual physical location within the chip's
12 area of all the transistors that make up the cells. The physical design process also will
13 designate the actual routes of the wires that will connect the cells on the IC. Timing
14 analysis also can be performed in the physical design phase. The result of physical design
15 is a detailed layout that is used to fabricate the IC.

16 61. A fundamental problem in EDA is the separation of logic synthesis
17 and physical design. The logic synthesis phase produces a netlist that is used to drive
18 physical design. This netlist is the result of choices and optimizations made without
19 complete knowledge of the physical placement and routing of cells. This can produce a
20 final design that is significantly less than optimal. For example, a design that appears to
21 satisfy the timing requirements for the IC during the logic synthesis phase may not satisfy
22 those requirements once the physical design process is completed and actual electrical
23 paths are determined. Such a result may require a repetition of the logic synthesis phase
24 to create a new netlist. Iterating between logic synthesis and physical design in a
25 repetitive search for a satisfactory result may consume significant time and may never
26 produce a solution close to the optimal layout. For these reasons, integrating the steps of
27 logic synthesis and physical design has long been a goal of EDA tool developers.

28

1 **DR. LUKAS VAN GINNEKEN**

2 62. Lukas van Ginneken, a luminary in the EDA field, graduated cum
3 laude in electrical engineering from Eindhoven University of Technology in the
4 Netherlands in 1984. He received a Ph.D. degree from Eindhoven University in electrical
5 engineering in 1989. Dr. van Ginneken's Ph.D. dissertation relates to the field of physical
6 design, and in particular to the application of stepwise refinement to layout design. In this
7 work, he presented automatic algorithms to solve various physical design problems. Dr.
8 van Ginneken has authored or co-authored numerous research papers on logic synthesis
9 and physical design, and he has been granted several patents in the EDA field.

10 63. From 1989 to 1995, Dr. van Ginneken worked at IBM's T.J. Watson
11 Research Center in Yorktown Heights, New York, and at IBM's Somerset Design Center
12 in Austin, Texas. During his tenure at IBM, Dr. van Ginneken worked on the problem of
13 integrating logic synthesis with physical design as well as improving optimizations
14 within logic synthesis and physical design. For example, Dr. van Ginneken co-authored
15 the paper "Timing Verification and Optimization for the PowerPC Processor Family,"
16 published in the Proceedings of the International Conference on Computer Design in
17 October 1994. This paper discusses a timing optimizer and describes a method for
18 automatically deriving timing constraints.

19 64. While at IBM, Dr. van Ginneken also developed a fundamental
20 algorithm for the placement of buffers in the pathways between cells. This work is
21 presented in the article "Buffer Placement in Distributed RC-tree Networks for Minimal
22 Elmore Delay," published in the Proceedings of the International Symposium on Circuits
23 and Systems, May 1990, and is widely known today simply as "van Ginneken's
24 algorithm."

25 65. By the time Dr. van Ginneken left IBM, his work in the area of
26 physical design and logic synthesis included the following papers:
27
28

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

- (a) "Efficient orthonormality testing for synthesis with pass transistor selectors," by M.R.C.M. Berkelaar and L. van Ginneken, accepted at the International Workshop on Logic Synthesis, June 1995.
- (b) "In the driver's seat of BooleDozer," by D. Brand and R.F. Damiano, L. van Ginneken, A.D. Drumm, in Proc. Int. Conf. on Computer Design, pp. 518-521, Boston, Oct. 10-12, 1994.
- (c) "Grammar-based optimization of synthesis scenarios," by A. Kuehlmann and L. van Ginneken, in: Proc. Int. Conf. on Computer Design, pp. 20-25, Boston, Oct. 10-12, 1994.
- (d) "Tuning of logic synthesis scenarios," by L. van Ginneken and A. Kuehlmann, Workshop Notes of the Int. Workshop on logic synthesis, paper P7c, Tahoe City, May 23-26, 1993.
- (e) "Fanin ordering in multi-slot timing," by L. van Ginneken, Proc. Int. Conf. on Computer Design, pp. 44-47, Cambridge, Oct. 11-14, 1992.
- (f) "The complexity of adaptive annealing," by R.H.J.M. Otten and L. van Ginneken, Proc. Int. Conf. on Computer Design, pp. 404-407, Cambridge, Sept. 17-19, 1990.
- (g) "Optimal slicing of plane point placements," by L. van Ginneken and R.H.J.M. Otten, Proc. European Design Automation Conf., pp. 322-336, Glasgow, March 12-15, 1990.
- (h) "The annealing algorithm," by R.H.J.M. Otten and L. van Ginneken, ISBN 07923-9022-9, Boston: Kluwer, 1989.
- (i) "The predictor-adaptor paradigm – automation of custom layout by flexible design," by L. van Ginneken, Ph.D. thesis, ISBN 90-9002703-3, Eindhoven, 1989.
- (j) "Doubly folded transistor matrix layout," by L. van Ginneken and J.T.J. van Eijndhoven, A.H.C.M. Brouwers, Digest Int. Conf. on Computer Aided Design, Santa Clara, Nov. 7-10, 1988.
- (k) "Stop criteria in simulated annealing," by R.H.J.M. Otten and L. van Ginneken, Proc. Int. Conf. on Computer Design, pp. 549-552, Port Chester, Oct. 3-5, 1988.
- (l) "An inner loop criterion for simulated annealing," by L. van Ginneken and R.H.J.M. Otten, Physics letters A, 130:429-435, 1988.

- 1 (m) "Soft Macro Cell generation by two dimensional
2 folding," by L. van Ginneken and J.T.J. van
3 Eijndhoven, P.R.M. van Teeffelen, T.J. Deckers, Proc.
4 Int. Symp. on Circuits and Systems, pp. 727-730,
5 Espoo, June 1988.
- 6 (n) "Gridless routing of general floor plans," by L. van
7 Ginneken and J.A.G. Jess, Digest Int. Conf. on
8 Computer Aided Design, pp. 30-33, Santa Clara, Nov.
9 9-12, 1987.
- 10 (o) "Wire planning for stackable designs," by R.K.
11 Brayton, C.L. Chen, J.A.G. Jess, R.H.J.M. Otten and L.
12 van Ginneken, Proc. Int. Symp. on VLSI technology,
13 pp. 269-273, Taipeh, May 13-15, 1987.
- 14 (p) "Global wiring for custom layout design," by L. van
15 Ginneken and R.H.J.M. Otten, Proc. Int. Symp. on
16 Circuits and Systems, pp. 207-208, Kyoto, June 5-7,
17 1985.
- 18 (q) "Floor plan design using simulated annealing," by
19 R.H.J.M. Otten and L. van Ginneken, Digest Int. Conf.
20 on Computer Aided Design, pp. 96-98, Santa Clara,
21 Nov, 1984.
- 22 (r) "Stepwise layout refinement," by L. van Ginneken and
23 R.H.J.M. Otten, Proc. Int. Conf. on Computer Design, pp.
24 30-36, Port Chester, Oct. 8-11, 1984.

25 66. In June of 1995, Dr. van Ginneken left IBM to join Synopsys. As the
26 foregoing list of papers reflects, when Dr. van Ginneken joined Synopsys he already
27 possessed a high degree of knowledge, skill, and expertise in logic synthesis, physical
28 design, and the integration of logic synthesis with physical design. At Synopsys, Dr. van
29 Ginneken continued to work on many of the problems and techniques that were the focus
30 of his research at IBM, including the integration of logic synthesis with physical design.
31 Synopsys benefited not only from Dr. van Ginneken's talents, but also from the
32 knowledge and experience he had gained at IBM.

33 THE IBM-SYNOPSYS JOINT DEVELOPMENT AGREEMENT

34 67. When Dr. van Ginneken joined Synopsys in 1995, IBM and
35 Synopsys were entering into a joint technology development agreement relating to EDA
36 ("the IBM-Synopsys Agreement"). Under the IBM-Synopsys Agreement, any inventions
37
38

1 that resulted from the work performed thereunder became the joint property of Synopsys
2 and IBM.

3 68. The Synopsys engineers involved in this joint project included Dr.
4 van Ginneken, Narendra Shenoy, Robert Damiano, Tony Ma, and Mahesh Iyer. The IBM
5 engineers involved in this joint project included Prabhakar Kudva, Leon Stok, Tony
6 Drumm, and Andrew Sullivan.

7 69. On July 1, 1997, Synopsys filed a patent application based on this
8 joint project. The patent application named Dr. van Ginneken and Narendra Shenoy as
9 inventors. No IBM engineer was named on the patent application, despite the fact that
10 one or more IBM engineers also contributed in a significant way to the subject matter of
11 one or more of the patent claims. This application matured into the '114 Patent, issued
12 April 23, 2002. Like the application, the issued '114 Patent names only Dr. van Ginneken
13 and Narendra Shenoy as inventors.

14 THE FOUNDING OF MAGMA

15 70. Lukas van Ginneken left Synopsys and joined Magma as one of
16 several founders in May 1997. A central goal of the new company was to create advanced
17 EDA software that effectively integrated logic synthesis with physical design. In addition
18 to Dr. van Ginneken, Magma's founders included:

19 (a) *Rajeev Madhavan*. Before co-founding Magma and becoming
20 its President and CEO, Mr. Madhavan already had been an entrepreneur in the EDA
21 industry. He had founded and served as the President and CEO of Ambit Design Systems,
22 Inc. ("Ambit"), the first credible competitor to Synopsys in logic synthesis, and had co-
23 founded LogicVision, a BIST supplier. Mr. Madhavan also had worked at Cadence
24 Design Systems, Inc. ("Cadence"), a leading EDA company. At Cadence, he led the
25 invention and development of the Verilog-A product.

26 (b) *Hamid Savoj*. Dr. Savoj, a renowned expert and innovator in
27 logic optimization, joined Magma in May 1997 as Principal Engineer. Dr. Savoj holds a
28 Ph.D. in electrical engineering and computer science (focusing on computer aided design

1 of VLSI) from the University of California, Berkeley. Before joining Magma, Dr. Savoj
2 was a senior member of the consulting staff at Cadence, where he developed state-of-the-
3 art algorithms for area and performance optimization of logic circuits. Dr. Savoj also has
4 co-authored numerous papers relating to logic synthesis.

5 (c) *Karen Vahtra*. Ms. Vahtra had worked for Synopsys and
6 Ambit before joining Magma. An expert in the application of static timing analysis, Ms.
7 Vahtra had co-authored papers on the integration of logic synthesis and physical design
8 before joining Magma.

9 71. In addition to the founders, Magma assembled a talented group of
10 engineers known for their expertise in EDA and related fields. The engineers included:

11 (a) *Premal Buch*. When he joined Magma, Dr. Buch was a Ph.D.
12 candidate in electrical engineering at the University of California, Berkeley. He had
13 extensive research experience in logic synthesis and had worked for Cadence.

14 (b) *Hardy Kwok-Shing Leung*. When Hardy Leung joined Magma
15 he was a Ph.D. candidate in computer science (focusing on VLSI computer-aided design)
16 at the University of California, Los Angeles. He previously had worked for Cadence,
17 where he was a senior member of its technical staff and worked on global routing, clock
18 routing, wire-sizing, and buffer insertion. He has co-authored several papers on routing.

19 (c) *Hsiao-Ping Tseng*. When he joined Magma, Dr. Tseng was a
20 Ph.D. candidate in electrical engineering at the University of Washington, Seattle, and had
21 co-authored numerous papers in EDA-related fields.

22 (d) *Patrick Groeneveld*. Before joining Magma, he was an
23 associate professor of electrical engineering at Delft University of Technology in the
24 Netherlands and specialized in CAD for VLSI.

25 (e) *Joseph Hutt, Jr.* Before joining Magma, Mr. Hutt had worked
26 for over 20 years as an electrical engineer for IBM. His responsibilities at IBM included
27 serving as program director for VLSI Design Systems.

28

1 (f) *Timothy Burks*. When he joined Magma, Dr. Burks had
2 earned a Ph.D. in electrical engineering from the University of Michigan. He had worked
3 as an engineer for IBM. There, he was the architect and original developer of DeLTA
4 ("Device Level Timing Analysis"), a static transistor timing level analyzer for custom
5 CMOS circuits.

6 (g) *Hong Cai*. Dr. Cai, holder of a Ph.D. in electrical engineering
7 from Delft University of Technology, had worked for Synopsys as a senior member of its
8 technical staff before joining Magma. Dr. Cai also had authored or co-authored numerous
9 publications relating to IC routing.

10 (h) *Robert Swanson*. When Mr. Swanson joined Magma he had
11 almost ten years of IC design experience at IBM. He also has been granted several
12 semiconductor patents.

13 (i) *Raymond Nijssen*. When he joined Magma, Raymond
14 Nijseen held an master's degree in electrical engineering from Eindhoven University. He
15 has been granted several IC patents.

16 72. The Magma engineering team also included Michel R.C.M.
17 Berkelaar, Manjit Borah, Cornelius A.J. van Eijk, and Eduard P. Huijbregts, all of whom
18 hold Ph.D.'s.

19 73. Magma undertook rigorous measures to ensure that its engineers did
20 not use or disclose at Magma any trade secret or other proprietary information derived
21 from their work at former employers. To this end, all Magma employees, including Dr.
22 van Ginneken, were required to execute a Proprietary Information and Inventions
23 Agreement. This agreement includes the following provision:

24 During my employment by [Magma] I will not improperly use
25 or disclose any confidential information or trade secrets, if
26 any, of any former employer or any other person to whom I
27 have an obligation of confidentiality, and I will not bring onto
28 the premises of [Magma] any unpublished documents or any
property belonging to any former employer or any other
person to whom I have an obligation of confidentiality unless
consented to in writing by that former employer or person.

1 74. Magma took further precautions, including periodically archiving all
2 its source code. This archiving has continued to present.

3 75. Magma also retained outside counsel, Orrick Herrington & Sutcliffe
4 LLP ("Orrick"), to perform intellectual property due diligence at Magma in late 1998 and
5 early 1999. As part of this effort, Orrick engaged Dr. Marios Papaefthymiou to analyze
6 the provenance of Magma's source code. Dr. Papaefthymiou holds a Ph.D. in Electrical
7 Engineering and Computer Science from the Massachusetts Institute of Technology. At
8 the time of the Magma due diligence, he was an Assistant Professor in the Department of
9 Electrical Engineering and Computer Science at the University of Michigan.

10 76. As part of the due diligence, Dr. Papaefthymiou and Orrick attorneys
11 interviewed the developers of Magma's code and confirmed that the developers had not
12 brought any confidential information to Magma from any third party. In addition, Dr.
13 Papaefthymiou reviewed Magma's source code and interviewed its developers to ensure
14 that it had been developed independently at Magma without the use or incorporation of
15 any third-party intellectual property. Based on this due diligence, Orrick concluded there
16 was no reason to believe that Magma had used or incorporated any intellectual property of
17 third parties.

18 **DEVELOPMENT OF THE MAGMA PATENTS**

19 77. At Magma, Dr. van Ginneken conceived of the inventions disclosed
20 and claimed in the Magma Patents. Dr. van Ginneken did not use any proprietary
21 information or trade secrets of Synopsys in creating those inventions, consistent with Dr.
22 van Ginneken's execution of the Proprietary Information and Inventions Agreement with
23 Magma. Instead, the inventions were improvements and extensions of matters already in
24 the public domain.

25 78. The inventions disclosed in the Magma Patents include novel
26 applications of the concepts of "constant delay" and "logical effort." Delay refers to the
27 time it takes for a cell to carry out its function and to communicate its result to the next
28 cell. As the demand or "load" on a cell increases, the delay increases. Under the concept

1 of "constant delay," however, each cell is modeled as having a delay that does not change
2 with changes in load. As the design of an IC proceeds through various stages, increases in
3 a cell's load imposed by changes in the design are accommodated by increasing the cell
4 size to provide more power so that the delay remains constant. The "logical effort"
5 concept refers to a formulation of gate delay as a function of three factors: (a) logical
6 effort, which does not depend on the size of the cell; (b) electrical effort (or gain); and (c)
7 parasitic delay.

8 79. The concept of constant delay has existed in the public domain since
9 at least 1995. In particular, the concept of constant delay for use in logic synthesis is
10 discussed in the article, "A Delay Model for Logic Synthesis of Continuously-Sized
11 Networks," by J. Grodstein et al., from Digest Int. Conf. On Computer Aided Design, pp.
12 458-462, San Jose, California November 5-9, 1995 ("the Grodstein article"). The
13 Grodstein article presents the basic concept of holding a cell's delay constant while its
14 area is adjusted to accommodate changes in load.

15 80. The concept of constant delay is also explored in a companion to the
16 Grodstein article entitled, "Logic Decomposition During Technology Mapping," by Eric
17 Lehman, Yosinori Watanabe, Joel Grodstein and Heather Harkness, from Proceedings of
18 the 1995 IEEE/ACM international conference on Computer-aided design, pp. 264-271
19 ("the Lehman article"). The Lehman article addresses the problem of mapping a set of
20 logical expressions onto library cells. The Lehman article describes achieving more
21 optimal mapping by using technology-dependent features of the library.

22 81. The concept of logical effort was introduced in "Logical Effort:
23 Designing for Speed on the Back of an Envelope," by Ivan E. Sutherland and Robert F.
24 Sproull, from Proceedings of the 1991 University of California/Santa Cruz conference on
25 Advanced research in VLSI, p.1-16, April 1991 ("the Sutherland article"). The
26 Sutherland article separates logical effort, which expresses the logical complexity of the
27 gate, from electrical effort, which expresses the gain of the gate. The Sutherland article
28 uses the logical effort formulation to approximate a relationship among area, load, and

1 delay.

2 82. The logical effort concept is extended in "Generalized Delay
3 Optimization of Resistive Interconnections Through an Extension of Logical Effort," by
4 Kumar Venkat, from Proceedings of ISCAS 1993, pp. 2106-2109 ("the Venkat paper").
5 The Venkat paper describes an extension of the logical effort concept that accommodates
6 the resistance of wires in addition to their capacitance.

7 83. In creating the inventions disclosed in the Magma Patents, Dr. van
8 Ginneken drew from the extensive work available in the public domain, including the
9 publications listed above, and relied on his background and experience in EDA. The
10 novel aspects of the inventions were conceived entirely at Magma.

11 84. On December 24, 1997, Magma filed with the PTO the provisional
12 patent application that ultimately resulted in the Magma Patents.

13 85. On September 17, 2002, the PTO issued the '446 Patent, entitled
14 "Timing Closure Methodology." Dr. van Ginneken is named as the sole inventor and
15 Magma is the assignee.

16 86. On April 26, 2004, the PTO issued the '438 Patent, entitled "Timing
17 Closure Methodology." Dr. van Ginneken is named as the sole inventor and Magma is
18 the assignee.

19 **SYNOPSIS' LACK OF INTEREST IN MAGMA'S TECHNOLOGY**

20 87. Through the contributions of Dr. van Ginneken and other members of
21 its engineering staff, Magma developed the concept of fixed timing. The fixed timing
22 methodology implements a constant delay model within an automatic tool that integrates
23 timing and placement into a single-pass design flow from RTL specifications to layout.
24 This methodology establishes and optimizes circuit speeds prior to physical design.
25 During physical design, the circuit design is refined to achieve a final timing that is very
26 close to the circuit speed previously established. Magma became the first EDA company
27 to offer this integrated approach.

1 88. The fixed timing approach eliminates the timing iterations that exist
2 in conventional design flows, and thus can significantly reduce the time it takes to design
3 and produce deep submicron integrated circuits. Given the importance of time-to-market
4 in the semiconductor industry, EDA software that accelerates the IC design process can
5 provide a significant competitive advantage to chip designers. This technology has
6 enabled Magma to make competitive inroads against companies such as Synopsys.

7 89. That Magma's software employs a fixed timing methodology was no
8 secret to Synopsys, because Magma repeatedly discussed the concept with Synopsys. For
9 example, in February 1998, representatives of Synopsys met with representatives of
10 Magma to explore the possibility of Magma being merged into or acquired by Synopsys.
11 At the meeting, Magma informed Synopsys that it was developing a fixed timing
12 methodology. In response, Synopsys asserted that Magma's approach would not work.

13 90. Later in 1998, Synopsys and Magma representatives met again. At
14 the meeting, Magma showed its fixed timing design methodology to Synopsys. Once
15 again, Synopsys was not interested in Magma's technology. Instead, Synopsys' Chairman
16 and Chief Executive Officer Aart de Geus began telling investment analysts that Magma's
17 fixed timing technology was a failure.

18 91. In the summer of 2001, Magma made a presentation about its
19 technology at a meeting sponsored by investment bank Credit Suisse First Boston. Senior
20 Synopsys management, including Dr. de Geus, attended. Magma's presentation featured
21 its fixed timing methodology as central to its proprietary technology. Synopsys yet again
22 expressed skepticism about Magma's approach: Dr. de Geus argued that fixed timing did
23 not work.

24 92. On November 20, 2001, Magma announced its initial public offering.
25 Magma stressed the importance of its fixed timing methodology to its products:
26 "Magma's proprietary FixedTiming® methodology and single data model architecture are
27 the technical foundation for Magma's Blast Fusion and Blast Chip products. The
28 FixedTiming methodology allows Magma's products to reduce the timing closure

1 iterations that are often required between the front-end and back-end processes in
2 conventional integrated circuit design flows. The single data model contains all of the
3 logical and physical information about the chip design.”

4 **THE IBM-MAGMA PATENT LICENSE**

5 93. On March 24, 2004, Magma and IBM entered into a patent license
6 agreement. Under this license agreement, Magma is broadly licensed to all patents owned
7 by IBM that were filed before a specified date.

8 94. As explained above, by operation of law and pursuant to the IBM-
9 Synopsys Agreement, IBM is an owner of the ‘114 Patent. Thus, Magma is licensed to
10 the ‘114 Patent pursuant to the Magma-IBM patent license agreement.

11 **SYNOPSYS’ CLAIMS AGAINST MAGMA**

12 95. On July 1, 2004, Magma wrote to Synopsys, requesting that
13 Synopsys confirm whether certain Magma patents (including the two Magma Patents at
14 issue here) were applicable to Synopsys’ gain-based delay model or any other Synopsys
15 design solution. Over two months passed with no word from Synopsys.

16 96. On September 17, 2004, Synopsys finally responded by filing this
17 lawsuit, which alleges that Magma itself infringes the Magma Patents as well as the ‘114
18 Patent.

19 97. Magma does not infringe the ‘114 Patent because the ‘114 Patent’s
20 claims are fundamentally different from the innovative technology underlying Magma’s
21 products. Among other reasons that Magma does not infringe this patent, Magma’s
22 products, unlike the requirements of every claim of the ‘114 Patent, do not “establish[] a
23 convergence criterion based on a partition size.” Magma’s single-pass approach also
24 distinguishes its technology from the iterative approaches of the ‘114 Patent.

25 98. Moreover, the work that led to the development of the inventions
26 claimed in the ‘114 Patent was part of the joint project between IBM and Synopsys to
27 which IBM engineers made significant contributions. By operation of law and pursuant to
28 the IBM-Synopsys Agreement, IBM is a co-owner of the ‘114 Patent. Because IBM is a

1 co-owner of the '114 Patent, Synopsys' failure to name IBM as a plaintiff in this suit is
2 fatal to Synopsys' claim for infringement of the '114 Patent. Magma is also licensed to
3 the '114 Patent and therefore cannot be liable for infringement of that patent as a matter of
4 law.

5 99. Synopsys also cannot assert the Magma Patents against Magma. As
6 explained above, Dr. van Ginneken conceived of the inventions claimed in the Magma
7 Patents at Magma, not at Synopsys. Thus, Magma – not Synopsys – owns the Magma
8 Patents.

9 100. In the alternative, if Synopsys could somehow establish that Dr. van
10 Ginneken conceived the inventions disclosed in the Magma Patents while he was at
11 Synopsys, Magma could not be liable for infringing the patents as a matter of law. If Dr.
12 van Ginneken developed the inventions at Synopsys, that work would have occurred as
13 part of the joint project between IBM and Synopsys to which IBM engineers made
14 significant contributions. Thus, by operation of law and pursuant to the IBM-Synopsys
15 Agreement, IBM would be an owner of the Magma Patents. Accordingly, Synopsys could
16 not assert the patents against Magma without naming IBM as a plaintiff, and Magma
17 would be licensed under them pursuant to its patent license agreement with IBM. Thus,
18 Magma cannot be liable for infringing the Magma Patents.

19 **SYNOPSIS' FALSE STATEMENTS AND UNFAIR COMPETITION**

20 101. Synopsys' misconduct is not limited to filing this baseless lawsuit.
21 Synopsys is engaging in a campaign with the press and with Magma's customers and
22 competitors to spread false and misleading statements about Magma and its products.

23 102. On the day it filed this lawsuit, Synopsys issued a press release
24 stating that "After a thorough review, Synopsys has determined that it is not infringing the
25 cited patents, and further determined Synopsys rightfully owns the two van Ginneken
26 patents. Accordingly, Synopsys today filed suit in Federal court against Magma under the
27 van Ginneken patents to enforce its rights as the owner of the inventions and to bar
28 Magma from practicing Synopsys' technologies."

1 103. These and other false statements by Synopsys about Magma and its
2 technology have begun to negatively affect Magma's relationships with its customers and
3 its reputation in the marketplace. Synopsys has informed customers that Magma has
4 stolen trade secrets and that Synopsys owns the technology which underlies Magma's
5 products. In response, Magma has had to make significant and extraordinary efforts to
6 maintain customer relationships as a result of the uncertainty and doubt that Synopsys'
7 statements have created in the market. Magma has had to visit customers to correct
8 Synopsys' false statements and persuade the customers not to take their business
9 elsewhere despite Synopsys' false statements.

10 **FIRST COUNTERCLAIM FOR RELIEF**

11 **(NON-INFRINGEMENT OF THE '114 PATENT)**

12 104. Magma incorporates by reference the allegations set forth in the
13 previous paragraphs.

14 105. On April 23, 2002, the PTO issued the '114 Patent, entitled "Method
15 for the Physical Placement of an Integrated Circuit Adaptive to Netlist Changes," upon an
16 application filed in the names of Narendra Shenoy and Lukas van Ginneken.

17 106. Synopsys claims to be the owner of the '114 Patent.

18 107. There exists an actual and justiciable controversy within the meaning
19 of 28 U.S.C. §§ 2201 and 2202 between Magma and Synopsys with respect to the
20 inventorship, ownership, validity, enforceability, and infringement of the '114 Patent and
21 Magma's alleged liability for infringement thereof.

22 108. Magma does not infringe, or contribute to or induce the infringement
23 of, the '114 Patent.

24 **SECOND COUNTERCLAIM FOR RELIEF**

25 **(INVALIDITY OF THE '114 PATENT)**

26 109. Magma incorporates by reference the allegations set forth in the
27 previous paragraphs.

1 110. On April 23, 2002, the PTO issued the '114 Patent, entitled "Method
2 for the Physical Placement of an Integrated Circuit Adaptive to Netlist Changes," upon an
3 application filed in the names of Narendra Shenoy and Lukas van Ginneken.

4 111. Synopsys claims to be the owner of the '114 Patent.

5 112. There exists an actual and justiciable controversy within the meaning
6 of 28 U.S.C. §§ 2201 and 2202 between Magma and Synopsys with respect to the
7 inventorship, ownership, validity, enforceability, and infringement of the '114 Patent and
8 Magma's alleged liability for infringement thereof.

9 113. The '114 Patent is invalid because it fails to satisfy the conditions for
10 patentability specified in Title 35 of the United States Code.

11 **THIRD COUNTERCLAIM FOR RELIEF**

12 **(IBM'S JOINT OWNERSHIP OF THE '114 PATENT)**

13 114. Magma incorporates by reference the allegations set forth in the
14 previous paragraphs.

15 115. On April 23, 2002, the PTO issued the '114 Patent, entitled "Method
16 for the Physical Placement of an Integrated Circuit Adaptive to Netlist Changes," upon an
17 application filed in the names of Narendra Shenoy and Lukas van Ginneken.

18 116. Synopsys claims to be the owner of the '114 Patent.

19 117. There exists an actual and justiciable controversy within the meaning
20 of 28 U.S.C. §§ 2201 and 2202 between Magma and Synopsys with respect to the
21 inventorship, ownership, validity, enforceability, and infringement of the '114 Patent and
22 Magma's alleged liability for infringement thereof.

23 118. IBM is a joint owner of the '114 Patent.

24 **FOURTH COUNTERCLAIM FOR RELIEF**

25 **(NO LIABILITY FOR INFRINGEMENT**

26 **OF THE '114 PATENT DUE TO LICENSE)**

27 119. Magma incorporates by reference the allegations set forth in the
28 previous paragraphs.

120. On April 23, 2002, the PTO issued the '114 Patent, entitled "Method for the Physical Placement of an Integrated Circuit Adaptive to Netlist Changes," upon an application filed in the names of Narendra Shenoy and Lukas van Ginneken.

121. Synopsys claims to be the owner of the '114 Patent.

122. There exists an actual and justiciable controversy within the meaning of 28 U.S.C. §§ 2201 and 2202 between Magma and Synopsys with respect to the inventorship, ownership, validity, enforceability, and infringement of the '114 Patent and Magma's alleged liability for infringement thereof.

123. Magma cannot be liable for infringing the '114 Patent because Magma is licensed under the '114 Patent.

FIFTH COUNTERCLAIM FOR RELIEF
(OWNERSHIP OF THE MAGMA PATENTS)

124. Magma incorporates by reference the allegations set forth in the previous paragraphs.

125. Magma holds record title to and is the legal and equitable owner of all right, title and interest in and to the '446 and '438 Patents.

126. Notwithstanding that Magma is the owner of all right, title and interest in and to the '446 and '438 Patents, in its Complaint, Synopsys claims to be the sole owner of all of the inventions claimed in the '446 Patent and the '438 Patent. Synopsys also is claiming to the public that Synopsys, rather than Magma, is the true owner of the '446 and '438 Patents.

127. There is a substantial, actual and continuing controversy between Magma and Synopsys as to the ownership of the '446 Patent and the '438 Patent.

128. Synopsys' false claims of ownership in the '446 Patent and the '438 Patent have harmed Magma and will continue to harm Magma until such time as Synopsys is enjoined from making such claims.

129. Pursuant to the Federal Declaratory Judgment Act, Magma requests the Court declare that Synopsys has no ownership right in either the '446 Patent or the

1 '438 Patent and that the Court further declare Magma the owner of all right, title and
2 interest in and to the '446 Patent and the '438 Patent.

3 **SIXTH COUNTERCLAIM FOR RELIEF**

4 **(NON-INFRINGEMENT OF THE '446 PATENT)**

5 130. Magma incorporates by reference the allegations set forth in the
6 previous paragraphs.

7 131. On September 17, 2002, the PTO issued to Magma the '446 Patent,
8 entitled "Timing Closure Methodology," upon an application filed in the name of Lukas
9 P. P. P. van Ginneken.

10 132. Synopsys claims to be the owner of the '446 Patent.

11 133. There exists an actual and justiciable controversy within the meaning
12 of 28 U.S.C. §§ 2201 and 2202 between Magma and Synopsys with respect to the
13 inventorship, ownership, validity, enforceability, and infringement of the '446 Patent and
14 Magma's alleged liability for infringement thereof.

15 134. If Magma does not own the '446 Patent, Magma does not infringe, or
16 contribute to or induce the infringement of, the '446 Patent.

17 **SEVENTH COUNTERCLAIM FOR RELIEF**

18 **(IBM'S JOINT OWNERSHIP OF THE '446 PATENT)**

19 135. Magma incorporates by reference the allegations set forth in the
20 previous paragraphs.

21 136. On September 17, 2002, the PTO issued to Magma the '446 Patent,
22 entitled "Timing Closure Methodology," upon an application filed in the name of Lukas
23 P. P. P. van Ginneken.

24 137. Synopsys claims to be the owner of the '446 Patent.

25 138. There exists an actual and justiciable controversy within the meaning
26 of 28 U.S.C. §§ 2201 and 2202 between Magma and Synopsys with respect to the
27 inventorship, ownership, validity, enforceability, and infringement of the '446 Patent and
28 Magma's alleged liability for infringement thereof.

1 139. If Magma does not exclusively own the '446 Patent, IBM is a joint
2 owner of the '446 Patent.

3 **EIGHTH COUNTERCLAIM FOR RELIEF**
4 **(NO LIABILITY FOR INFRINGEMENT**
5 **OF THE '446 PATENT DUE TO LICENSE)**

6 140. Magma incorporates by reference the allegations set forth in the
7 previous paragraphs.

8 141. On September 17, 2002, the PTO issued to Magma the '446 Patent,
9 entitled "Timing Closure Methodology," upon an application filed in the name of Lukas
10 P. P. P. van Ginneken.

11 142. Synopsys claims to be the owner of the '446 Patent.

12 143. There exists an actual and justiciable controversy within the meaning
13 of 28 U.S.C. §§ 2201 and 2202 between Magma and Synopsys with respect to the
14 inventorship, ownership, validity, enforceability, and infringement of the '446 Patent and
15 Magma's alleged liability for infringement thereof.

16 144. If Magma does not own '446 Patent, Magma cannot infringe the '446
17 Patent because Magma is licensed under the '446 Patent.

18 **NINTH COUNTERCLAIM FOR RELIEF**
19 **(INVALIDITY OF THE '446 PATENT)**

20 145. Magma incorporates by reference the allegations set forth in the
21 previous paragraphs.

22 146. On September 17, 2002, the PTO issued to Magma the '446 Patent,
23 entitled "Timing Closure Methodology," upon an application filed in the name of Lukas
24 P. P. P. van Ginneken.

25 147. Synopsys claims to be the owner of the '446 Patent.

26 148. There exists an actual and justiciable controversy within the meaning
27 of 28 U.S.C. §§ 2201 and 2202 between Magma and Synopsys with respect to the
28

1 inventorship, ownership, validity, enforceability, and infringement of the '446 Patent and
2 Magma's alleged liability for infringement thereof.

3 149. If Magma does not own the '446 Patent, the '446 Patent is invalid
4 because it fails to satisfy the conditions for patentability specified in Title 35 of the United
5 States Code.

6 **TENTH COUNTERCLAIM FOR RELIEF**
7 **(NON-INFRINGEMENT OF THE '438 PATENT)**

8 150. Magma incorporates by reference the allegations set forth in the
9 previous paragraphs.

10 151. On April 20, 2004, the PTO issued to Magma the '438 Patent,
11 entitled "Timing Closure Methodology," upon an application filed in the name of Lukas
12 P. P. P. van Ginneken.

13 152. Synopsys claims to be the owner of the '438 Patent.

14 153. There exists an actual and justiciable controversy within the meaning
15 of 28 U.S.C. §§ 2201 and 2202 between Magma and Synopsys with respect to the
16 inventorship, ownership, validity, enforceability, and infringement of the '438 Patent and
17 Magma's alleged liability for infringement thereof.

18 154. If Magma does not own the '438 Patent, Magma does not infringe, or
19 contribute to or induce the infringement of, the '438 Patent.

20 **ELEVENTH COUNTERCLAIM FOR RELIEF**
21 **(IBM'S JOINT OWNERSHIP OF THE '438 PATENT)**

22 155. Magma incorporates by reference the allegations set forth in the
23 previous paragraphs.

24 156. On April 20, 2004, the PTO issued to Magma the '438 Patent,
25 entitled "Timing Closure Methodology," upon an application filed in the name of Lukas
26 P. P. P. van Ginneken.

27 157. Synopsys claims to be the owner of the '438 Patent.

1 158. There exists an actual and justiciable controversy within the meaning
2 of 28 U.S.C. §§ 2201 and 2202 between Magma and Synopsys with respect to the
3 inventorship, ownership, validity, enforceability, and infringement of the '438 Patent and
4 Magma's alleged liability for infringement thereof.

5 159. If Magma does not exclusively own the '438 Patent, IBM is a joint
6 owner of the '438 Patent.

7 **TWELFTH COUNTERCLAIM FOR RELIEF**
8 **(NO LIABILITY FOR INFRINGEMENT**
9 **OF THE '438 PATENT DUE TO LICENSE)**

10 160. Magma incorporates by reference the allegations set forth in the
11 previous paragraphs.

12 161. On April 20, 2004, the PTO issued to Magma the '438 Patent,
13 entitled "Timing Closure Methodology," upon an application filed in the name of Lukas
14 P. P. P. van Ginneken.

15 162. Synopsys claims to be the owner of the '438 Patent.

16 163. There exists an actual and justiciable controversy within the meaning
17 of 28 U.S.C. §§ 2201 and 2202 between Magma and Synopsys with respect to the
18 inventorship, ownership, validity, enforceability, and infringement of the '438 Patent and
19 Magma's alleged liability for infringement thereof.

20 164. If Magma does not own the '438 Patent, Magma cannot be liable for
21 infringing the '438 Patent because Magma is licensed under the '438 Patent.

22 **THIRTEENTH COUNTERCLAIM FOR RELIEF**
23 **(INVALIDITY OF THE '438 PATENT)**

24 165. Magma incorporates by reference the allegations set forth in the
25 previous paragraphs.

26 166. On April 20, 2004, the PTO issued to Magma the '438 Patent,
27 entitled "Timing Closure Methodology," upon an application filed in the name of Lukas
28 P. P. P. van Ginneken.

1 167. Synopsys claims to be the owner of the '438 Patent.

2 168. There exists an actual and justiciable controversy within the meaning
3 of 28 U.S.C. §§ 2201 and 2202 between Magma and Synopsys with respect to the
4 inventorship, ownership, validity, enforceability, and infringement of the '438 Patent and
5 Magma's alleged liability for infringement thereof.

6 169. If Magma does not own the '438 Patent, the '438 Patent is invalid
7 because it fails to satisfy the conditions for patentability specified in Title 35 of the United
8 States Code.

9 **FOURTEENTH COUNTERCLAIM FOR RELIEF**
10 **(UNFAIR COMPETITION IN VIOLATION OF**
11 **CAL. BUS. & PROF. CODE § 17200 ET SEQ.)**

12 170. Magma incorporates by reference the allegations set forth in the
13 previous paragraphs.

14 171. By reason of the foregoing, Synopsys has been, and is, engaged in
15 "unlawful, unfair or fraudulent business practices" in violation of California Business and
16 Professions Code §§ 17200 *et seq.*, and in acts of unfair competition in violation of the
17 common law.

18 172. Synopsys' acts complained of herein have damaged and will continue
19 to damage Magma irreparably. Magma has no adequate remedy at law for such wrongs
20 and injuries. The damage to Magma includes harm to its goodwill and reputation that
21 money cannot compensate. Magma is therefore entitled to preliminary and permanent
22 injunctions restraining and enjoining Synopsys and its agents, servants, employees,
23 representatives, successors and assigns, and those acting in concert with them or on their
24 behalf, from making false and misleading statements that Synopsys owns the Magma
25 Patents and that Magma infringes the '114 Patent and the Magma Patents.

26
27
28

PRAYER FOR RELIEF

WHEREFORE, Defendant and Counterclaimant Magma prays:

(1) that the Court dismiss with prejudice the Complaint of plaintiff Synopsys, that Synopsys take nothing by reason of the Complaint, and that judgment be rendered in favor of Magma;

(2) that the Court render judgment declaring that Magma has not infringed and is not infringing the '114 Patent;

(3) that the Court render judgment declaring that IBM is a joint owner of the '114 Patent;

(4) that the Court render judgment declaring that Magma cannot be liable for infringing the '114 Patent because Magma is licensed under the '114 Patent;

(5) that the Court render judgment declaring that the '114 Patent is invalid;

(6) that the Court render judgment declaring that Synopsys has no ownership interest whatsoever in the '446 Patent or in the '438 Patent;

(7) that the Court render judgment re-affirming and declaring that Magma is the owner of all right, title and interest in and to the '446 and '438 Patents;

(8) that Synopsys, its agents, servants, employees, representatives, successors and assigns, and those acting in privity or in concert with them or on their behalf, be preliminarily and permanently enjoined from claiming or otherwise stating that (a) Synopsys is the owner, in whole or in part, of the '446 or '438 Patents, or any inventions claimed therein, or (b) Magma infringes the '114 Patent, the '446 Patent, or the '438 Patent;

(9) that if Magma does not own the '446 Patent, the Court render judgment declaring that Magma has not infringed and is not infringing the '446 Patent;

(10) that if Magma does not exclusively own the '446 Patent, the Court render judgment declaring that IBM is a joint owner of the '446 Patent;

1 (11) that if Magma does not own the '446 Patent, the Court render
2 judgment declaring that Magma cannot be liable for infringing the '446 Patent because
3 Magma is licensed under the '446 Patent;

4 (12) that if Magma does not own the '446 Patent, the Court render
5 judgment declaring that the '446 Patent is invalid;

6 (13) that if Magma does not own the '438 Patent, the Court render
7 judgment declaring that Magma has not infringed and is not infringing the '438 Patent;

8 (14) that if Magma does not exclusively own the '438 Patent, the Court
9 render judgment declaring that IBM is a joint owner of the '438 Patent;

10 (15) that if Magma does not own the '438 Patent, the Court render
11 judgment declaring that Magma cannot be liable for infringing the '438 Patent because
12 Magma is licensed under the '438 Patent;

13 (16) that if Magma does not own the '438 Patent, the Court render
14 judgment declaring that the '438 Patent is invalid;

15 (17) that the Court render judgment declaring this to be an exceptional
16 case under 35 U.S.C. § 285;

17 (18) that Magma be awarded its attorneys' fees and costs; and

18 (19) that Magma be awarded such other and further relief as the Court
19 deems proper.

20 Dated: October 21, 2004

21 GEORGE A. RILEY
22 CHRISTOPHER D. CATALANO
23 RYAN K. YAGURA
24 LUANN L. SIMMONS
25 O'MELVENY & MYERS LLP

26 By /s/ George A. Riley
27 George A. Riley

28 Attorneys for Defendant and
Counterclaimant MAGMA DESIGN
AUTOMATION, INC.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
2
2
2
2
2
2
2
2
2
2

Dated: October 21, 2004

By /s/ George A. Riley
George A. Riley

Attorneys for Defendant and
Counterclaimant MAGMA DESIGN
AUTOMATION, INC.

1 **CERTIFICATION OF INTERESTED ENTITIES OR PERSONS**

2 Pursuant to Civil L.R. 3-16, the undersigned certifies that the following listed
3 persons, associations of persons, firms, partnerships, corporations (including parent
4 corporations) or other entities (i) have a financial interest in the subject matter in
5 controversy or in a party to the proceeding, or (ii) have a non-financial interest in that
6 subject matter or in a party that could be substantially affected by the outcome of the
7 proceeding:

8 IBM's contract or property rights may be affected by the outcome of this
9 proceeding.

10
11 Dated: October 21, 2004

12 GEORGE A. RILEY
13 RYAN K. YAGURA
14 CHRISTOPHER D. CATALANO
15 LUANN L. SIMMONS
16 O'MELVENY & MYERS LLP

17 By /s/ George A. Riley
18 George A. Riley

19 Attorneys for Defendant and
20 Counterclaimant MAGMA DESIGN
21 AUTOMATION, INC.

22
23
24
25
26
27
28
SF1:564434.1

EV 314044804US 28 January 2005

PTO/SB/82 (09-04)

Approved for use through 11/30/2005. OMB 0651-0035

U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

ed to respond to a collection of information unless it displays a valid OMB control number.

Under:

EV314044804US

REVOCATION OF POWER OF ATTORNEY WITH
NEW POWER OF ATTORNEY
AND
CHANGE OF CORRESPONDENCE ADDRESS

Application Number	09/054,379 (Pat. 6,453,446)
Filing Date	2 April 1998 (Issued: 17 Sep 2002)
First Named Inventor	van Ginneken
Art Unit	unknown
Examiner Name	Matthew Smith
Attorney Docket Number	SYNP 103

I hereby revoke all previous powers of attorney given in the above-identified application.

☐ A Power of Attorney is submitted herewith.

OR

☒ I hereby appoint the practitioners associated with the Customer Number:

36454

☒ Please change the correspondence address for the above-identified application to:

☒ The address associated with
Customer Number:

36454

OR

☐ Firm or
Individual Name

Address

City

State

Zip

Country

Telephone

Fax

I am the:

☐ Applicant/Inventor.

☒ Assignee of record of the entire interest. See 37 CFR 3.71.
Statement under 37 CFR 3.73(b) is enclosed. (Form PTO/SB/96)

SIGNATURE of Applicant or Assignee of Record

Signature *Deirdre Hanford*

Name Deirdre Hanford

Date 28 January 2005

Telephone (650) 584-4201

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below.

☐ *Total of _____ forms are submitted.

This collection of information is required by 37 CFR 1.36. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

STATEMENT UNDER 37 CFR 3.73(b)Applicant/Patent Owner: van GinnekenApplication No./Patent No.: 09/054,379/6,453,446 Filed/Issue Date: 2 April 1998/17 September 2002Entitled: TIMING CLOSURE METHODOLOGYSYNOPSIS, INC., a CORPORATION

(Name of Assignee)

(Type of Assignee, e.g., corporation, partnership, university, government agency, etc.)

states that it is:

1. ☒ the assignee of the entire right, title, and interest; or
2. ☐ an assignee of less than the entire right, title and interest.
The extent (by percentage) of its ownership interest is _____ %

in the patent application/patent identified above by virtue of either:

A. ☐ An assignment from the inventor(s) of the patent application/patent identified above. The assignment was recorded in the United States Patent and Trademark Office at Reel _____, Frame _____, or for which a copy thereof is attached.

OR

B. ☒ A chain of title from the inventor(s), of the patent application/patent identified above, to the current assignee as shown below:

1. From: Lukas P.P. van Ginneken To: Synopsis, Inc.
The document was recorded in the United States Patent and Trademark Office at Reel _____, Frame _____, or for which a copy thereof is attached.
2. From: _____ To: _____
The document was recorded in the United States Patent and Trademark Office at Reel _____, Frame _____, or for which a copy thereof is attached.
3. From: _____ To: _____
The document was recorded in the United States Patent and Trademark Office at Reel _____, Frame _____, or for which a copy thereof is attached.

☒ Additional documents in the chain of title are listed on a supplemental sheet.

☒ Copies of assignments or other documents in the chain of title are attached.

[NOTE: A separate copy (i.e., a true copy of the original assignment document(s)) must be submitted to Assignment Division in accordance with 37 CFR Part 3, if the assignment is to be recorded in the records of the USPTO. See MPEP 302.08]

The undersigned (whose title is supplied below) is authorized to act on behalf of the assignee.

Deirdre Hanford
Signature

28 January 2005

Date

Deirdre Hanford

Printed or Typed Name

(650) 584-4201

Telephone Number

Sr. Vice President

Title

This collection of information is required by 37 CFR 3.73(b). The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Application No. 09/054,379
Patent 6,453,446

SUPPLEMENTAL SHEET TO STATEMENT UNDER 37 CFR 3.73(b)

Attached in support of the chain of title of the referenced application are copies of the following documents:

- 1) a copy of a Proprietary Information and Inventions Agreement attached hereto, signed by the inventor, that assigns to Synopsys, Inc. all inventions made, conceived, reduced to practice, or developed during the inventor's employment with Synopsys, Inc.;
- 2) a copy of a declaration by Robert Damiano, attached hereto, in which Robert Damiano attests that he received a draft patent application from the inventor ("Draft") during the inventor's employment with Synopsys, Inc., including as an attachment the email in which Robert Damiano received the Draft;
- 3) a copy of a chart, attached hereto, highlighting the common language shared between the Draft and the specification of U.S. Patent No. 6,453,446, from which U.S. Patent No. 6,725,438 claims priority as a continuation, from which U.S. Patent Application No. 10/828,547 claims priority as a continuing application, including as an attachment the Draft with page numbers referenced by the chart; and
- 4) a copy of a Complaint for Patent Infringement attached hereto for purposes of disclosure.

PROPRIETARY INFORMATION
AND
INVENTIONS AGREEMENT

The following confirms an agreement between me and Synopsys, Inc. (the "Company"), which is a material part of the consideration for my employment by the Company.

1. I recognize that the Company is engaged in a continuous program of research, development and production respecting its business, present and future, including fields generally related to its business and that the Company possesses and continues to possess information that has been created, discovered, developed or otherwise become known to the Company (including, without limitation, information created, discovered or developed by, or made known to, me during the period of or arising out of my employment by the Company) and/or in which property rights have been assigned, licensed or otherwise conveyed to the Company, which information has commercial value in the business in which the Company is engaged. All of the aforementioned information is hereinafter called "Proprietary Information." By way of illustration, but not limitation, Proprietary Information includes trade secrets, processes, data and know-how, computer software, improvements, inventions, works of authorship, techniques, marketing plans, strategies, forecasts and copyrightable material and customer lists.

2. I understand that my employment creates a relationship of confidence and trust between me and the Company with respect to any information:

(i) applicable to the business of the Company; or

(ii) applicable to the business of any client or customer of the Company, which may be known to me by the Company or by any client or customer of the Company, or learned by me during the period of my employment.

3. In consideration of my employment by the Company and the compensation received by me from the Company from time to time, I hereby agree as follows:

A. All Proprietary Information shall be the sole property of the Company and its assigns, and the Company and its assigns shall be sole owner of all patents, copyrights and other rights in connection therewith. I hereby assign to the Company any rights I may have or acquire in such Proprietary Information. At all times, both during my employment by the Company and after its termination, I will keep in confidence and trust all Proprietary Information, and I will not use or disclose any Proprietary Information or anything relating to it without the written consent of the Company, except as may be necessary in the ordinary course of performing my duties to the Company.

B. All documents, records, apparatus, equipment and other physical property, whether or not pertaining to Proprietary Information, furnished to me by the Company or produced by me or others in connection with my employment shall be and remain the sole property of the Company and shall be returned to the Company immediately as and when requested by the Company. Even if the Company does not so request, I shall return and deliver all such property upon termination of my employment by me or the Company for any reason and I will not take with me any such property or any reproduction of such property upon such termination.

C. I will promptly disclose to the Company, or any persons designated by it, all improvements, inventions, works of authorship, processes, techniques, know-how, formulae,

ATTORNEYS AND
CONSULTANTS ONLY

SY000003

data, ideas and other information (including, without limitation, my algorithms or software), whether or not patentable, made or conceived or reduced to practice or learned by me, either alone or jointly with others, during the term of my employment (all said improvements, inventions, works of authorship, processes, techniques, know-how, formulae, data, ideas and other information shall be hereinafter collectively called "Inventions").

D. I agree that all Inventions which I make, conceive, reduce to practice or develop (in whole or in part, either alone or jointly with others) during my employment shall be the sole property of the Company to the maximum extent permitted by Section 2870 of the California Labor Code (hereinafter called "Section 2870"), a copy of which is attached hereto as Exhibit B, and to the extent permitted by law, shall be "works made for hire." The Company shall be the sole owner of all patents, copyrights, trade secret rights, rights with respect to other intellectual property or other rights in connection therewith (including, without limitation, such rights in algorithms or software). I hereby assign to the Company any rights I may have or acquire in such Inventions. I agree to perform, during and after my employment, all acts deemed necessary or desirable by the Company to permit and assist it, at the Company's expense, in obtaining, maintaining and enforcing patents, copyrights, trade secret rights, rights with respect to such Inventions and/or other Inventions I have or may at any time assign to the Company in any and all countries. Such acts may include, but are not limited to, execution of documents and assistance or cooperation in legal proceedings. I hereby irrevocably designate and appoint the Company and its duly authorized officers and agents as my agents and attorneys-in-fact to act for and on my behalf and instead of me, to execute and file any applications or related filings and to do all other lawfully permitted acts to further the prosecution, maintenance and enforcement, issuance of patents, copyrights, trade secret rights, rights with respect to mask works or other rights thereon with the same legal force and effect as if executed by me.

E. As a matter of record, I attach hereto a complete list of all inventions or improvements relevant to the subject matter of my employment by the Company which have been made by me or jointly with others prior to my employment with the Company that I desire to remove from the operation of this Agreement, and I covenant that such list is complete. If no such list is attached to this Agreement, I represent that I have no such inventions and improvements at the time of signing this Agreement.

F. During the term of my employment and for one (1) year thereafter, I will not encourage or solicit any employee of the Company to leave the Company for any reason or devote less than all of any such employee's efforts to the affairs of the Company, provided that the foregoing shall not affect any responsibility I may have as an employee if the Company with respect to the bona fide hiring and firing of Company personnel.

G. I represent that my performance of all the terms of this Agreement will not breach any agreement to keep in confidence proprietary information acquired by me in confidence or in trust prior to my employment by the Company. I have not entered into, and I agree I will not enter into, any agreement, either written or oral, in conflict herewith.

H. I represent that execution of this Agreement, my employment with the Company and my performance of my proposed duties to the Company in the development of its business will not violate any obligations I may have to my former employer.

I. This Agreement does not require assignment of an invention which an employee cannot be obligated to assign under Section 2870. However, I will disclose any Inventions as required by Section 3(c) hereof regardless of whether I believe the Invention is protected by Section 2870, in order to permit the Company to engage in a review process to determine such issues as may arise. Such disclosure shall be received in confidence by the Company.

SY000004

**ATTORNEYS AND
CONSULTANTS ONLY**

4. This Agreement shall be effective as of the first day of my employment by the Company: June 26 1995 (date).

5. This Agreement shall be binding upon me, my heirs, executors, assigns and administrators and shall inure to the benefit of the Company, its successors and assigns.

Dated: 5/17/95




Signature

Lukas van Ginneken

(Print or Type Name)

Accepted and agreed to:

Synopsys, Inc.

By  _____

ATTORNEYS AND
CONSULTANTS ONLY

SY000005

EXHIBIT A
TO
SYNOPSIS, INC. PROPRIETARY INFORMATION
AND INVENTIONS AGREEMENT

1. The following is a complete list of all inventions or improvements relevant to the subject matter of my employment by Synopsis, Inc. (the "Company") that have been made or conceived or first reduced to practice by me alone or jointly with others prior to my employment by the Company that I desire to remove from the operation of the Company's Proprietary Information and Inventions Agreement.

_____ No inventions or improvements

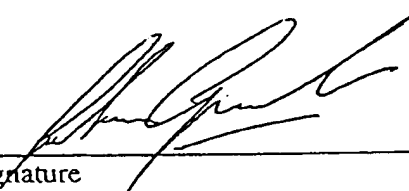
_____ See below

X Additional sheets attached

2. I propose to bring to my employment the following materials and documents of a former employer:

X No materials or documents

_____ See below



Signature

Lukas van Glnneken

Print or Type Name

ATTORNEYS AND
CONSULTANTS ONLY

SY000006

EXHIBIT B

TO

SYNOPSYS, INC. PROPRIETARY INFORMATION
AND INVENTIONS AGREEMENT

Section 2870. Application of provision providing that employee shall assign or offer to assign rights in inventions to employer.

(a) Any provision in an employment agreement which provides that an employee shall assign, or offer to assign, any of his or her rights in an invention to her or her employer shall not apply to an invention that the employee entirely on his or her own time without using the employer's equipment, supplies, facilities or trade secret information except for those inventions that either:

(1) Relate at the time of conception or reduction to practice of the invention to the employer's business, or actual or demonstrably anticipated research or development by the employer;

(2) result from any work performed by the employee for the employer.

(b) To the extent a provision in an employment agreement purports to require an employee to assign an invention otherwise excluded from being required to be assigned under subdivision (a), the provision is against the public policy of this state and is unenforceable.

ATTORNEYS AND
CONSULTANTS ONLY

SY000007

The following is a complete list of all inventions or improvements relevant to the subject matter of my employment by Synopsys, Inc. (the Company) that have been made or conceived or first reduced to practice by me alone or jointly with others prior to my employment by the Company that I desire to remove from the operation of the Company's Proprietary Information and Inventions Agreement.

[0] "Efficient orthonormality testing for synthesis with pass transistor selectors" by M. R. C. M. Berkelaar and -, accepted at the International Workshop on Logic Synthesis, June 1995.

[1] "Timing Verification and Optimization for the PowerPC Processor Family", by R.E. Mains, T. A. Mosher, - and R.F. Damiano, in: Proc. Int. Conf. on Computer Design, pp.390-393, Boston, Oct. 10-12, 1994.

[2] "In the driver's seat of BooleDozer" by D. Brand and R.F. Damiano, -, A. D. Drumm, in: Proc. Int. Conf. on Computer Design, pp. 518-521, Boston, Oct. 10-12, 1994.

[3] "Grammar-based optimization of synthesis scenarios" by A. Kuehlmann and -, in: Proc. Int. Conf. on Computer Design, pp. 20-25 Boston, Oct. 10-12, 1994.

[4] "Tuning of logic synthesis scenarios" by - and A. Kuehlmann, Workshop notes of the Int. Workshop on logic synthesis, paper P7c, Tahoe City, May 23-26, 1993.

[5] "Fanin ordering in multi-slot timing" by -, Proc. Int. Conf. on Computer Design, pp. 44-47, Cambridge, Oct. 11-14, 1992.

[6] "The complexity of adaptive annealing" by R. H. J. M. Otten and -, Proc. Int. Conf. on Computer Design, pp. 404-407, Cambridge, Sept. 17-19, 1990.

[7] "Buffer placement in distributed RC-tree networks for minimal Elmore delay" by -, Proc. Int. Symp. on Circuits and Systems, pp. 865-868, New Orleans, May 2-5, 1990.

[8] "Optimal slicing of plane point placements" by - and R. H. J. M. Otten, Proc. European Design Automation Conf. pp. 322-236, Glasgow, March 12-15, 1990.

[9] The annealing algorithm by R. H. J. M. Otten and -, ISBN 07923-9022-9, Boston:Kluwer, 1989.

[10] The predictor-adaptor paradigm - automation of custom layout by flexible design by -, Ph.D. thesis, ISBN 90-9002703-3, Eindhoven, 1989.

[11] "Doubly folded transistor matrix layout" by - and J. T. J. van Eijndhoven, A. H. C. M. Brouwers, Digest Int. Conf. on Computer Aided Design, Santa Clara, Nov. 7-10, 1988.

[12] "Stop criteria in simulated annealing" R. H. J. M. Otten and -, Proc.

Int. Conf. on Computer Design, pp.549-552, Port Chester, Oct. 3-5, 1988.

[13] "An inner loop criterion for simulated annealing" by - and R.H.J.M. Otten, Physics letters A, 130:429-435, 1988.

[14] "Soft Macro Cell generation by two dimensional folding" by - and J. T. J. van Eijndhoven, P. R. M. van Teeffelen, T. J. Deckers, Proc. Int. Symp. on Circuits and Systems, pp. 727-730, Espoo, June 1988.

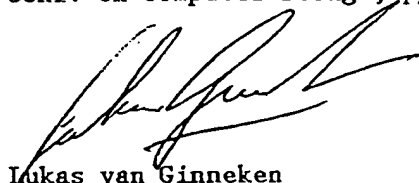
[15] "Gridless routing of general floor plans" by - and J. A. G. Jess, Digest Int. Conf. on Computer Aided Design, pp. 30-33, Santa Clara Nov. 9-12, 1987.

[16] "Wire planning for stackable designs", by R. K. Brayton, C. L. Chen, J. A. G. Jess, R. H. J. M. Otten and -, Proc. Int. Symp. on VLSI technology, pp.269-273, Taipeh, May 13-15, 1987.

[17] "Global wiring for custom layout design" by - and R. H. J. M. Otten, Proc. Int. Symp. on Circuits and Systems. pp.207-208, Kyoto, June 5-7, 1985.

[18] "Floor plan design using simulated annealing" by R. H. J. M. Otten and -, Digest Int. Conf. on Computer Aided Design, pp. 96-98, Santa Clara, Nov. 1984.

[19] "Stepwise layout refinement" by - and R. H. J. M. Otten, Proc. Int. Conf. on Computer Design, pp. 30-36, Port Chester, Oct.8-11, 1984.



Lukas van Ginneken

ATTORNEYS AND
CONSULTANTS ONLY

SY000009

DECLARATION OF ROBERT DAMIANO

I, Robert Damiano, declare as follows:

1. The following statements are based on my personal knowledge. If called upon to testify, I could and would competently testify as to the matters set forth herein.
2. I am an employee of Synopsys, Inc. My present position is Vice-President of the Advanced Technology Group. In September 1996, my position at Synopsys was Director in the Advanced Technology Group. Lukas van Ginneken, who was also employed at Synopsys during that time, worked on one of my projects.
3. On or about September 9, 1996, I received an email from Lukas van Ginneken that included a draft patent application. A true and correct copy of such email is attached hereto as Exhibit A.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of Declarant:

Robert Damiano

Declarant's Signature:

Robert Damiano

Date:

28 January 2005

Exhibit A

[Document Follows]

TO: robertd@synopsys.com
From: Lukas van Ginneken <lukas@synopsys.com>
Subject: patent
Date: 1996-09-09 22:07:54 GMT

SYNOPSYS CONFIDENTIAL

APPLICATION FOR UNITED STATES PATENT

in the name of

LUKAS PAUL PIETER PEPIJN VAN GINNEKEN

of

SYNOPSYS, INC.

for

METHOD FOR ACHIEVING TIMING CLOSURE OF DIGITAL NETWORKS

AND

METHOD FOR AREA OPTIMIZATION OF DIGITAL NETWORKS UNDER TIMING CLOSURE

class 364/489

BACKGROUND OF THE INVENTION

This application relates to a method for achieving timing closure of digital networks consisting of structuring and mapping and a method for area optimization of digital networks using placement and sizing, while maintaining timing closure.

(Prior Art)

Figure 1 shows the conventional approach to digital network synthesis.

Digital network synthesis is a process in which computer programs optimize digital networks. At the beginning of the synthesis process, a human designer 110 specifies a design 111 at a high level of abstraction using a high level design specification language, such as Verilog or VHDL.

In step 101 the abstract network specification of the design is transformed into an unmapped digital network representation in memory by a computer program, such as HDL Compiler, available from Synopsys Inc. of Mountain View, California.

In step 102 Logic synthesis algorithms optimize the network by changing the structure of the network without changing the function of the network.

In step 103, logic network synthesis algorithms map the abstract network representation to cells from the library.

Step 104 optimizes the size of the cells. The size of the cell, together with the load of the cell determines the delay of the cell. The sizing algorithm adjusts the sizes of the cells. Changing the sizes of the cells affects delay and area, and the sizing algorithm manipulates the cell sizes so as to minimize delay and area.

Steps 102, 103 and 104 are performed by a computer program, such as "Design Compiler" TM available from Synopsys Inc., of Mountain View, California.

Step 105 determines the placement of the cells on the chip. Placement algorithms attempt to keep the length of the nets short, as longer nets need more area on the chip and the increased net load of longer nets will make the network slower. The network remains unchanged during the placement.

Finally step 106 determines the exact routing of the nets on the chip. Steps 105 and 106 are done by a computer program, such as "Cell Ensemble" TM available from Cadence Inc. of San Jose, California.

(Problems with prior art)

The major problem with the conventional approach is that the net length and hence the cell delay is not known until after placement. Before placement, net length must be estimated. This is usually done with an estimation function or table which gives the load of a net based on its fanout. Experience has shown that it is very difficult to estimate the length of the nets accurately. Essentially net length behaves as a random variable.

The result is unpleasant surprises after placement step 105. Some nets turn out to be longer than expected, and because of the longer delays the timing constraints are not met. Timing closure is not certain until after step 105.

If timing closure is not achieved the options the designer has are expensive and unreliable. He may choose to fix the design manually, which is difficult and time consuming, because the automatically optimized network is hard to understand. He may choose to change his HDL specification and repeat the synthesis process. Again timing closure will not be certain until after placement, which means that the entire process needs to be traversed before the designer knows if his HDL changes were successful.

A common method of dealing with inaccurate net load estimates is to use net load estimates which are considerably larger than accurate estimates. This causes the sizes of the cells to be considerably larger than necessary but reduces the probability of not meeting the timing constraints after placement. Clearly using cells with sizes which are larger than necessary is wasteful in both silicon area and power consumption. The chips thus synthesized will be larger, cost more to produce and use more electrical power than necessary.

A second problem with the conventional approach is that the effect

of synthesis decisions is hard to calculate. Performing timing analysis during optimization is very time consuming, and accounts for most of the run time of conventional synthesis systems.

In step 103 it is difficult to take decisions based on delay without knowing load and size of the cells as well.

In step 104, changing the size of a cell affects the loads of the fanin cells, and thus the delay of the fanin cells. In more complex delay models, which take into account the transition time of the signals, also the delay of the fanout cells is affected. Usually the size parameter cannot have any arbitrary value. Because the library of cells has been designed before the network synthesis started, only a few sizes are available. 3 or 4 sizes per cell is common. This makes it harder to find a good solution.

In step 105 the placement program will modify the net lengths. Depending on which location was chosen for each cell, the length of each net can be different. As the length differs, the capacitive load of the net changes, and as a result, the delay of the cell driving the net changes. Therefore the delays which were carefully optimized during the logic synthesis, are very different after placement, and the optimization of the network is not very good.

Much of the progress in the state of the art can be characterized as increased integration. This is represented in figure 1 as various feedback paths, which repeat and alternate steps. The general direction has been towards programs which do structuring, mapping, sizing and placement simultaneously. It has lead to increasingly complex software systems which are slow and difficult to design and maintain.

Iterating between placement and sizing has been especially hard to execute because placement programs are not sold by the same design automation software vendors as logic synthesis programs. Also they are not run by the same users; the logic synthesis program is often run by the designer, who also wrote the HDL specification. The placement program is often run by the silicon chip manufacturer, after the design is considered complete.

SUMMARY OF THE INVENTION

(objectives)

It is an object of the present invention to achieve timing closure as quickly as possible in the synthesis process. This will give the human designer early warning if the design is infeasible, because it is over constrained.

The invention achieves this object by

- speeding up the algorithms in the synthesis process
- performing the feasibility check early in the synthesis process, that is, before placement, instead of after placement.
- maintaining feasibility throughout the remainder of the synthesis process, so that it can be guaranteed to succeed and can be executed automatically.

The present invention overcomes the problems of the conventional approach by not choosing a size for a cell at all. Rather than choosing a default size, as conventional methods do, we choose a delay and let the size implicitly be whatever it needs to be to meet that delay.

In the conventional method of optimization, the structure, mapping, size and placement are chosen to optimize delay and area. In our formulation of the problem, we choose the structure, mapping, delay and placement, to optimize size and area. In our formulation, size only affects the area, so area only remains as an optimization goal.

The present invention speeds up the programs by simplifying or eliminating timing analysis. While optimizing network delay in order to achieve timing closure, the delays are constant, which will speed up delay calculation. Also, since changes to the network do not change the delay of the cells, the amount of recalculation is drastically reduced. While optimizing area after timing closure has been achieved, timing analysis is not needed, as the delay of the cells does not change. The one step where timing analysis using complex delay models is necessary is in the stretching step. Here too, library design rules are taken into account.

The present invention maintains timing closure after it has been achieved by adjusting the size of the cell during or after placement. The adjustments compensate for the fact that the placement algorithm can assign different net lengths to different nets and that these lengths are difficult to predict before placement.

(language of the main claims)

In accordance with the purpose of this invention, as embodied and broadly described herein, the invention is a method for the structuring and mapping of an unmapped digital network comprising the following steps:

- a) structuring of the digital network, using network slack as an optimization goal, where network slack is calculated assuming that the delay of the cells of the network is constant with respect to load.
- b) mapping of the cells of the network, using network slack as an optimization goal, where network slack is calculated assuming that the delay of the cells of the network is constant with respect to load.
- c) estimation of the area of the network based on net load

In further accordance with the purpose of this invention, as embodied and broadly described herein, the invention is a method for the placement and sizing of cells of a mapped digital network, the method comprising the steps of:

- d) Choosing a target delay for each cell.
- e) Computing the network slack using the target delays.
- f) Placement of the cells of the network.
- g) Sizing of the cells of the network such that the network meets the network slack as computed by step b).

In further accordance with the purpose of this invention, as embodied and broadly described herein, the invention is a method for the placement of the cells of a digital network, the method comprising the steps, of:

- f1) The calculation of net weights that reflect the change of network area due to sizing as a function of net length.
- f2) Placement of the cells of the network where the weighted network net length is used as a placement objective, the weighted network net length being the sum of the weighted net lengths of all nets, each net length being

multiplied by a weight,

Objects and advantages of the invention will be set forth in part in the description which follows and in part will be obvious from the description or may be learned by practice of the invention. The objects and advantages of the invention will be realized and attained by means of the elements and combinations particularly pointed out in the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute part of this specification, illustrate several embodiments of the invention and, together with the description, serve to explain the principles of the invention.

Fig. 1 is a flow chart showing the flow of the conventional method.

Fig. 2 is a flow chart showing the flow according to the present invention.

Fig. 3 is a block diagram of a computer.

Fig. 4 is a schematic diagram of a digital network.

Fig. 5 is a schematic diagram of the electronic and the physical implementations of a cell.

Fig. 6 is a timing diagram illustrating the concept of slack and other timing concepts.

Fig. 7 is a graph showing the relationship between the delay of a cell, the size of a cell and the load of a cell.

Fig. 10 is a

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Reference will now be made in detail to the preferred embodiments of the invention, examples of which are illustrated in the accompanying drawings. Wherever possible, the same reference numbers will be used throughout the drawings to refer to the same or like parts.

(what insight led to this invention?)

(Software patent application)

The preferred embodiment of the present invention is as one or more computer programs. The digital network, its elements and attributes, exist during the process of the method only as data structures in the memory of the computer. Methods in prior art are known to persons of ordinary skill in the art to convert the design data in memory eventually to an actual physical implementation of the network.

(Computer system)

Figure 1 is a block diagram of a computer system 100 in accordance with

the present invention, Computer system 100 includes a central processing unit 101, bus 102, memory 103, input device 104 and output device 105. It will be understood by a person of ordinary skill in the art that computer system 100 can also include numerous elements not shown in the figure for the sake of clarity, such as disk drives, tape drives, mice, printers, network connections, additional CPUs, etc. Memory 103 contains a program 107, which embodies the invention, and a data structure representation of the network 106.

(Network terminology)

Figure 2 is a schematic diagram of a digital network. Digital "network" 200 is composed of a plurality of "cells" 205, 206, 207, 208, 209, connected by a plurality of "nets". Each cell (e.g. 208) has one or more inputs 212, 213, and a single output 214. Each net is connected to one output and one or more inputs. The cells can be combinational "gates" 207, 208, 209, whose function is represented as an expression in the Boolean algebra, using AND, OR and NOT operators, or the cells can be registers 205, 206. All feed back loops 210, 211 in the network contain at least one register. Cells whose outputs are connected to the inputs of a cell are called the fanin of the latter cell. Cells whose inputs are connected to the output of a cell are called the fanout of the latter cell.

(Computation)

The digital network performs a logic "function" by processing digital binary input data in a number of cycles. The input data is presented to the network on its "primary inputs" 201, 202, and the result of the computation of the network function is presented at the "primary outputs" 203, 204, of the network. The computation of the function takes one or more cycles. During each cycle the gate functions are calculated. The results are stored in the registers for use in the next cycle. The data flow is from inputs to outputs and will be assumed to be pictured as going from left to right throughout this text.

(Mapping)

Each cell can be "mapped" to a "book" in which case a electronic realization in terms of transistors has been chosen, and physical attributes such as delay are known. If a cell is not mapped to a book it is "unmapped", in which case no electronic realization has been selected, and the function of the cell is only known in abstract terms, such as Boolean algebraic expressions. A "library" of books is designed in advance, before synthesis starts. These books are of generic types, and can be used to build arbitrary designs.

(Size)

The cell has a delay, an area and its input pins have an input pin capacitance. The "size" of the cell is a multiplier which is applied to equally to all transistor channel widths in the electronic circuit of the cell. Thus size of a cell is a scale factor which is used to scale its output load driving capability (see below), its area, and its input pin loads.

(Timing constraints)

Each primary input or primary output has an associated delay,

called the **input delay**, respectively **output delay**, which represents delays external to the network. The network needs a certain amount of time to perform one cycle, called the **cycle delay**. Together the cycle delay, the input delays and the output delays form the timing constraints of the network. Meeting the timing constraints is called **timing closure** and it is a major objective of the synthesis process.

(Timing)

The delay of a path is measured as the sum of the gate delays over said path from begin point to end point. The cycle delay is the maximum of all path delays. Primary outputs and register inputs are timing path end points. Primary inputs and register outputs are timing path begin points.

(Slack)

The **arrival time** of the data at a gate is computed by taking the maximum arrival time of its fanin cells each increased by the delay from the input pin to the output pin. The arrival times can be computed by traversing the network from left to right, that is, from timing begin points to timing end points. Similarly the **required time** of the data at a gate can be computed by traversing the network from end points to begin points. The required time is the minimum required time of its fanout cells, each reduced by the delay from the input to the output pin of that fanout cell.

The difference between the required time and the arrival time is the **slack**. If the arrival time is smaller then the required time, the timing constraints are met, and the slack is positive. If the arrival time is larger then the required time, the timing constraints are not met, and the slack is negative. The arrival time and required time may be different depending on whether the data is zero (0) or one (1). There also may be multiple arrival times and multiple required times to model a variety of timing constraints. All slacks can be summarized as a single worst slack number, called the **network slack**. Timing closure is achieved if the network slack is non-negative.

(Delay model)

The delay D of a gate depends on many factors, among them its function, its size S and the capacitive load C of the gate. The delay may be different for different inputs of the gate and it may be different for the falling and the rising transition. It is important to note that the dependency on size and load can be captured as the dependency on a single parameter C/S , and the delay D is non-negative and monotonically increasing with C/S .

$$\begin{aligned} C &> 0 \\ S &> 0 \\ D &= f(C/S) \\ f(C/S) &> 0 \\ f'(C/S) &> 0 \end{aligned}$$

Figure 4 illustrates the relationship between the three variables, size S , delay D and capacitive load C . Each of the three planes shows the relationship between two variables, while the third variable is constant, (not necessarily zero).

(Detailed description)

The preferred embodiment of the invention is a software program that can be stored in the memory of a computer, and can be executed by the central processing unit of the computer so that the computer performs the method described herein.

The software program consists of many parts or subprograms which together perform the method described in this invention. The essence of the invention is that logic synthesis is done in a size independent way, and that sizes are determined after placement, and that we guarantee that that the delay numbers before placement can be met by sizing after placement.

The preferred embodiment consists of three parts: (See fig...) to wit

- a) The logic synthesis program
- b) The placement program
- c) The sizing program

- analyze the library
- read logic
- library independent optimization.

The first step is to analyze the library that will be used for logic synthesis. The library contains the cells that will be used to implement the logic function. Contrary to the standard method of performing logic synthesis, we will assume that each cell can be sized by a continuous, positive real variable S , which increases both the load driving capability of the cell and the area linearly. In other words, the area of a cell is $S \cdot A$ and the delay of a cell is $D = f(C/S)$.

The library analysis will determine a good value for C/S for each cell in the library. Using this value, it determines a constant delay for each gate.

(choice of C/S)

Since the library analysis is not dependent on the actual network being synthesized, library analysis can be performed before beginning the synthesis process. We will now continue to describe the actual automatic synthesis process, beginning at the with reading the design. The design is expressed in a high level design specification language, for example VHDL or Verilog, and is syntactically parsed and transformed into a logic network representation by a computer program, such as HDL Compiler, available from Synopsys Inc. of Mountain View, California.

(library independent optimization)

Initially the network is library independent and library independent optimizations are performed. Mostly these optimizations change the structure of the network, and the Boolean functions of the cells, without changing the overall function of the network. The types of optimizations that should be performed are behavioral optimizations, such as resource sharing, sequential optimizations such as retiming, algebraic optimizations such as kernel extraction and Boolean optimizations such as redundancy removal. There is a large amount of literature on how each of these classes of optimizations can be performed.

(mapping for delay)

Following the library independent optimisations the network is mapped to a library of cells. This means that the logic functions of the cells are implemented with actual cells from the library. During this process the a large body of literature exists already on the subject of mapping digital networks. The preferred embodiment would use a previously published algorithm such as

(post mapping optimizations)

Due to restrictions in run time, it is impractical to explore the entire design space during the mapping algorithm. Necessarily, the mapping algorithm has to ignore many possible solutions because either they are unlikely candidates or they are very similar (but not identical) to other, considered solutions. In addition, in the constant delay approach, it is easy to evaluate the impact on timing of synthesis decisions, but it is much harder to evaluate the impact on the total network area. Therefore the mapping algorithm necessarily cannot accurately optimize area.

(pin swapping)

An example of candidate mappings which are not explored during mapping because they are too similar to other mappings follows here: Often gates have several pins which are functionally interchangeable. For instance for a 4 input NAND gate, there are $4 \times 3 \times 2 \times 1$ possible permutations for the 4 input pins. Usually these pins are not equally fast, because of an inherent asymmetry of the electronic circuit. Because the differences are small, it is not worthwhile to consider all of these different of different permutations during mapping. It is more efficient to pick one arbitrarily and to select the best permutation of the inputs after mapping.

(structuring - boundary move)

Using constant delay it is considerably easier to predict the effect of a change to the network than with the conventional delay models. This can be used to do timing optimization by means of restructuring after technology mapping has been done. For example, we can use the "boundary move" transformation to reduce the number of levels in the logic in the mapped network. The boundary move transform, illustrated in fig x, reduces the number of levels by bringing connection x forward. To make the change legal it is necessary that gates x, y and z are fanout free. If not, they must be made fanout free by making a copy. In the conventional approach to logic synthesis, copying logic will increase the load on gates x, x, x and therefore increase the delay. To predict if the transformation will improve delay, or hurt delay, it was necessary to run a complete static timing analysis with accurate delay models. If the change actually worsened the delay, then the change would be undone.

In the constant delay model approach, the effect of this change can be easily predicted. Note that changes in loads do not affect delay. The only change that affects delay, is the change of the fanin of gate xxx. The delay can easily be predicted by simple addition of gate delays.

(area estimation)

To perform area optimization it is necessary to calculate the sizes of the cells. The sizes can be calculated in a straightforward manner from the loads. The loads are calculated by adding the net load and the pin load. The net load consists of the load of the net, which can be estimated using a conventional net load model, plus any other fixed load, such as the load of a primary output. The pin load is not fixed, that is, the load of an input pin depends on the size of the cell. This creates a dependency: To calculate the load of a cell, we need to calculate the size of its fanout cells. Therefore the algorithm starts calculating as far downstream as possible, and traverse the network in a direction opposite to the flow of data. In a combinational network this can be achieved by starting at the primary outputs and traversing the network in a levelized order towards the primary inputs. In a sequential network there may be one or more loops, resulting a cyclic dependency: there is no rightmost cell. In this case the computation can start anywhere in the cycle, and repeats the cycle several times, until the capacitances converge and the error is sufficiently small.

It is possible that this iteration will not converge and that the capacitance will increase in every iteration, by progressively larger amounts. This situation is detected by requiring the increment to be smaller than a preset maximum after a fixed number of iterations. The iteration does not converge if the network is an infeasible solution: The current network cannot be expected to work at this speed because its gain is too small. Changes need to be made to the network to increase the gain, which will usually mean increasing the delay of the network as well.

After the loads have been calculated the size can be calculated by dividing the actual load by the predetermined typical load. The input capacitance can be calculated by multiplying the unit gate input capacitance by the size. The ratio of these numbers is the size of the gate. The size is a scale factor, which can be applied to the area of the gate, to give the area of the sized gate. The area of the network can be estimated as the sum total of the areas of the sized gates, plus the net area as estimated from the total length of all nets.

(net weights)

Various algorithms, such as buffering and placement optimize the network by manipulating the loads in the network. Placement optimizes the net length, which directly related to the net load, and buffering reduces the load on a cell by adding extra delay. These algorithms can benefit from a more efficient calculation of the effect that changing the load of a cell has on sizing. We can do this by calculating single parameter per net, called the net weight, which represents the sensitivity of the total area of the network with respect to the load on that net.

This net weight can be calculated in a manner that is very similar to the calculation of the pin loads during the area calculation above. Starting at the primary inputs, the net weight of the first (left-most) gate is equal to its area per unit load. The net weights of the other cells can now be calculated with a recurrence relation traversing the network from left to right.....

to the calculation of the loads. Starting at the primary inputs the network to 0. Set the net load on the net in question to 1. Perform the iteration and calculate the area as described in the previous section.

Since all calculations are linear, the effects simultaneous changes in loads of several nets can be superimposed, that is, added together.

(buffering)

The next step in the synthesis process is the buffering step. The buffering algorithm adds buffers to the network guided by the timing analysis and the area analysis. In the constant delay model, the buffers have a fixed delay, and thus the impact of inserting a buffer is easily determined by subtracting the delay of the buffer from the slack. Thus the effect of adding a buffer on delay is always negative: a buffer always adds delay, never reduces delay. The main effect of adding a buffer is to save area because the source gate can be smaller because the it's load is smaller. The effect on area and gain can be determined by area analysis. Net weights.....

The buffering algorithm works as follows: First it finds locations in the network where a buffer can be added without increasing the network delay. This is done by subtracting the delay of the buffer from the local slack, to give the predicted slack after buffer insertion. If the predicted slack is larger than the network slack, then a buffer can be inserted without increasing the network delay. Next we have to calculate the reduction in load of this net, and check that area that is added by adding the buffer does not exceed the area saved by sizing down the source gate. The area added by inserting the buffer is simply the area of the buffer times its size, where the size is determined by the load on the buffer divided by the typical load of the buffer. The area saved by inserting the buffer can be calculated by first calculating the change in load due to the insertion of the buffer: some sinks are removed, the input load of the buffer is added, and the net load estimate may change as a result of the number of fanouts of the net changing. Using the net weight, we can estimate the impact on the network area. If the impact is positive (reduced area) the buffer is inserted. After the buffer is inserted, the capacitances need to be updated in the fanin cone of the buffer, while the net weights need to be updated in the fanout cone of the buffer.

(stretching)

The next step is the process of "Stretching" and "Compressing" the delays of the individual gates to meet the timing constraints. Gates which are on long paths which do not meet the delay constraint are "compressed" until the path does meet the timing constraint. Gates on the short paths which easily meet the timing constraints are "stretched". Gates with stretched delays require less area for the same load. In this step the delay of the gates is traded against the gain of the gates. When the delay of a gate decreases so does the gain of the gate. It is important that there is enough gain in the network.

Note that

this can be done entirely independent of the sizes of the gates. Sizes of gates are not determined until much later in the process.

The stretching algorithm has two phases. In the first phase it will compress the delays of cells on long paths to meet timing constraints. In the second phase it will stretch the delays of the

gates on short paths to save area. For the purpose of stretching and compressing registers can usually be considered to be part of a path which they originate, but not of a path that they terminate. The stretching algorithm considers the cells on a path by path basis, processing the path with the smallest slack first. The delay of each cell on the path is adjusted by an amount which is equal to the slack divided by the number of stages on the path. After a cell has been adjusted, it becomes "locked" and its delay cannot be changed by the stretching algorithm. Stages which are locked are not counted when calculating the adjustments. For the stretching phase, the algorithm continues as above. In this phase the delays of the cells are increased, not decreased. The path that we work on is not the worst path, but it is the worst path with a slack greater than 0. (All other paths now have a slack of 0).

(Rule based stretching)

(incremental rule based)

(placement)

In our process a conventional placement method is augmented to optimize the area of the placed network. All placement methods known work by gradual refinement of the placement. Periodically, during the placement process, we recalculate the estimated net lengths, using the most recent, accurate placement information. From the net lengths, it calculates the sizes of the cells in the network. (See area estimation). The updated sizes can then be used for further placement and for more accurate net length calculations.

Placement primarily manipulates the lengths of the nets. Using the net weights, the area of the network can efficiently be estimated.

(final or discrete sizing)

Other embodiments will be apparent to those skilled in the art from consideration of the specification and practice of the invention disclosed herein. It is intended that the specification and examples be considered as exemplary only, with a true scope of the invention being indicated by the following claims.

WHAT IS CLAIMED IS:

(CONSTANT DELAY SYNTHESIS)

1. A method for the structuring and mapping of an unmapped digital network comprising the following steps:

- a) structuring of the digital network, using network slack as an optimization goal, where network slack is calculated assuming that the delay of the cells of the network is constant with respect to load.
- b) mapping of the cells of the network, using network slack as an optimization goal, where network slack is calculated assuming that the delay of the cells of the network is constant with respect to load.
- c) estimation of the area of the network based on net load

(typical load)

3. The method of claim 1, the delay of a book is used for the delay of each cell, the delay of the book being chosen by choosing a load size ratio C/S for each book, which is independent of the network.

4. The method of claim 3, where the delay of a book is also determined by the choice of the input transition time, which is independent of the network.

(continuous buffering assumption)

5. The method of claim 3,

where a parameter C/S for each book

is chosen to have the largest

possible value such that a long chain of cells of identical books each cell in the chain having identical value of parameter C/S, said chain cannot have simultaneously improved delay and improved gain by adding a buffer at some point to the same chain, even when the parameter C/S is chosen optimally after adding the buffer.

(buffering)

6. The method of claim 1, with the additional step of buffer insertion before step c), the buffers being inserted on paths with positive slack, as determined by subtracting the delay of the buffer from the slack of the path.

(more buffering)

7. The method of claim 6, where the buffer is inserted if area is saved.

8. The method of claim 7, where the area savings are estimated using net weights which reflect the change of network area due to sizing as a function of net length.

9. The method of claim 8, where the calculation of the net weights

is performed by starting at the primary inputs and traversing the network in the direction of the data flow, while calculating the net weight of a cell by summing over the fanin of the cell the product of the net weight of the fanin cell divided by the gain plus the cells area/load sensitivity.

as $W_i = \sum_j W_j / g_{ij}$

(stretching)

10. The method of claim 1, where step a) is comprising of:

a1) choosing a delay for each book

a2) the delay of the cell assuming the delay of its book

a2) adjusting the delay of each cell based on the slack

11. The method of claim 10, where in step a2) the delay of each cell is adjusted equally among the stages which have the same slack.

12. The method of claim 10, where in step a2) the delay is adjusted on each path, such that the slack of each path becomes 0.

(globally optimal mapping)

13. The method of claim 1, where in step b) the mapping is performed in two steps:

b1) a traversal of the network from

primary inputs and registers from left to right, while choosing at each cell the fastest matching books from all available matching books, using the constant delays of the books and the fastest arrival times of the fanins

of the matching book.

b2) a traversal of the network from right to left, while choosing at each cell the fastest matching book from the candidates selected during the previous traversal.

(Area estimation)

14. The method of claim 1, where in step c) consists of the following steps:

- c1) estimation of the net length based on the number of fanout cells.
- c2) estimation of the capacitive load of the cells using the net length
- c3) calculation of the sizes from the capacitive load.
- c4) calculation of the network area by summation of the product of the book area times the cell size.

(sizing algorithm)

15. The method of claim 14, where step c2) is performed by starting at the primary outputs and traversing the network in the direction opposite to the data flow, while calculating the load of a cell, by summing over the fanout of the cell, the product of the load of the fanout cell divided by the gain plus the net load of the cell.

(iterate till convergence)

16. The method of claim 15, where the network has loops, the traversal is started at primary outputs and arbitrary cells in the loop, and the calculation is repeated until convergence.

17. The method of claim 16, where the traversal is started at primary outputs and registers.

(Area optimization)

18. The method of claim 1, where steps a) and b) also use network area as an optimization goal, in addition to network delay, the network area being estimated as in step c).

(retiming)

19. The method of claim 1, where the structuring step is preceded by a retiming step, where registers are moved in the network, while preserving the function of the network, and where change to the network do not affect the delay of the individual cells.

(I could produce a bunch more of these sort of claims)

(TIMING CLOSURE)

20. A method for the placement and sizing of cells of a mapped digital network, the method comprising the steps of:

- d) Choosing a target delay for each cell.
- e) Computing the network slack using the target delays.
- f) Placement of the cells of the network.
- g) Sizing of the cells of the network such that the network meets the network slack as computed by step b).

(sizing for better placement)

21. The method of claim 20, where step f) is performed in gradual steps, each step being followed by a sizing step g)

(partitioning)

22. The method of claim 21, where step f) is performed by repeated partitioning steps, partitioning the cells in the network into two or more groups, each group being assigned to an subdivision of the plane, alternating the partitioning steps with sizing steps, essentially similar to step g)

(iterative improvement)

23. The method of claim 21, where step f) is performed by choosing an arbitrary initial location in the two dimensional plane for each cell, the placement being optimized by repeatedly changing the location of one or two cells at a time, while performing a sizing step, essentially similar to step g) after each location change.

(sizing)

24. The method of claim 5, where step g) is consists of the following steps:
g1) calculation of the net length based on the available placement information
g2) calculation of the capacitive load of the cells using the net length
g3) calculation of the sizes from the capacitive load.

(sizing algorithm)

25. The method of claim 24, where step g2) is performed by starting at the primary outputs and traversing the network in the direction opposite to the data flow, while calculating the load of a cell, by summing over the fanout of the cell, the product of the load of the fanout cell divided by the gain plus the net load of the cell.

(iterate till convergence)

26. The method of claim 10, where the network has loops, the traversal is started at primary outputs and arbitrary cells in the loop, and the calculation is repeated until convergence.

27. The method of claim 26, where the traversal is started at primary outputs and registers.

(cell generation)

28. The method of claim 20, where following step g) the layout of the cells is generated automatically to yield the exact transistor sizes calculated by step g)

(discrete sizing)

29. The method of claim 24, where step g3) consists of selecting the most suitable size from a limited set of available sizes.

(Weighted Placement)

30. A method for the placement of the cells of a digital network, the method comprising the steps, of:
f1) The calculation of net weights that reflect the change of network area due to sizing as a function of net length.
f2) Placement of the cells of the network where the weighted network net length is used as a placement objective, the weighted network net length being the sum of the weighted net lengths of all nets, each net length being multiplied by a weight,

(Calculation of net weights)

31. The method of claim 30, where step f1) is performed by starting at the primary inputs and traversing the network in the direction of the data flow, while calculating the net weight of a cell by summing over the fanin of the cell the product of the net weight of the fanin cell divided by the gain plus the cells area/load sensitivity.
$$w_i = \sum_j w_j / g_{ij}$$

(iterate till convergence)

32. The method of claim 31, where the network has loops, the traversal is started at primary inputs and arbitrary cells in the loop, and the calculation is repeated until convergence.

33. The method of claim 32, where the traversal is started at primary inputs and registers.

(Weighted Placement for power)

34. A method for the placement of the cells of a digital network, the method comprising the steps, of:

f1) The calculation of net weights that reflect the change of network power due to sizing

as a function of net length.

f2) Placement of the cells of the network

where the weighted network net length is used as a placement objective, the weighted network net length being the sum of the weighted net lengths of all nets, each net length being multiplied by a weight,

(Calculation of net weights)

35. The method of claim 30, where step f1)

is performed by starting at the primary inputs and traversing the network in the direction of the data flow, while calculating the net weight of a cell by summing over the fanin of the cell the product of the net weight of the fanin cell divided by the gain plus the cells power/load sensitivity.

as $W_i = \sum_j W_j / g_{ij}$

(iterate till convergence)

36. The method of claim 31, where the network has loops, the traversal is started at primary inputs and arbitrary cells in the loop, and the calculation is repeated until convergence.

37. The method of claim 32, where the traversal is started at primary inputs and registers.

38. The method of claim 35, where the power/load sensitivity of a cell is calculated as the product of the switching frequency at that cell times the square of the voltage times the capacitance, where capacitance is calculated as the sum of the net load plus the internal capacitance of the cell, scaled with its size.

ABSTRACT

A method for the design of digital networks consisting of a plurality of cells. The invention uses constant delays during logic synthesis and sizes the cells after placement so as to meet the cycle delay predicted before synthesis. The method chooses a constant delay before logic synthesis and guarantees that it can maintain this delay after placement by means of sizing. Thus it overcomes the unpredictable effects of placement on the cycle delay. The invention also describes methods for choosing the sizes of the gates and a method for inserting buffers.

Synopsys Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 2</u></p> <p>The major problem with the <i>conventional approach</i> is that the net length and hence the cell delay is not known <i>until after placement</i>.</p>	<p><u>'446 PATENT AT 1:46-47</u></p> <p>Thus, under the <i>conventional</i> design approach, timing closure is not certain <i>until after placement</i>.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 2-3</u></p> <p>Before placement, <i>net length</i> must be <i>estimated</i>. This is usually done with <i>an estimation function or table which gives the load of a net based on its fanout</i>. Experience has shown that it is very difficult to <i>estimate</i> the length of the nets <i>accurately</i>.</p>	<p><u>'446 PATENT AT 1:37-40</u></p> <p>While <i>net lengths</i> have been <i>estimated</i> prior to placement by use of <i>an estimation function or table which gives the load value of a net based on</i> the number of <i>fanout</i> gates, this <i>estimation</i> function is usually <i>inaccurate</i>.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 3</u></p> <p>The result is unpleasant surprises <i>after placement</i> step 105. <i>Some nets turn out to be longer than expected</i>, and because of the <i>longer delays</i>, the <i>timing constraints</i> are not met. <i>Timing closure is not certain until after</i> step 105.</p>	<p><u>'446 PATENT AT 1:41:46</u></p> <p>This difficulty in accurately predicting net lengths leads to unpredictable delay effects <i>after cell placement</i> occurs. For example, <i>some nets turn out to be longer in length than expected</i>. These longer nets cause <i>longer delays</i> which prevent satisfaction of <i>timing constraints</i> in the digital circuit. Thus, under the conventional design approach, <i>timing closure is not certain until after</i> placement.</p>

¹ Note that page numbers do not appear on the original Draft Patent Application, but have been added for convenience. No other changes were made to the Draft Patent Application.

Synopsis Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 3</u></p> <p>If <i>timing closure</i> is not <i>achieved</i> the options the <i>designer</i> has are <i>expensive</i> and <i>unreliable</i>. He may choose to <i>fix the design manually, which is difficult and time consuming, because the automatically optimized network is hard to understand</i>. He may choose to <i>change his HDL specification and repeat the synthesis process</i>. Again <i>timing closure will not be certain until after placement</i>, which means that the entire <i>process</i> needs to be traversed <i>before the designer</i> knows if his <i>HDL changes were successful</i>.</p>	<p><u>'446 PATENT AT 1:48-60</u></p> <p>Failure to <i>achieve timing closure</i> after placement leads to additional <i>expenses</i> and other problems for the <i>designer</i>. To correct for failure to achieve timing closure, the <i>designer</i> has the option of <i>fixing the design manually, which is difficult and time consuming because the automatically optimized digital network is not easy to understand</i>. As a second option, the designer may <i>change the Hardware Description Language (HDL) specification and repeat the design process</i>. However, <i>timing closure will again not be certain until after placement</i>. Thus, the design <i>process</i> must again be repeated <i>before the designer</i> can determine if the <i>HDL specification changes were successful</i> in enabling timing closure.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 3</u></p> <p>A common method of dealing with <i>inaccurate net load estimates</i> is to use <i>net load estimates</i> which are <i>considerably larger than accurate estimates</i>. This <i>causes the sizes of the cells to be considerably larger than necessary</i> but reduces the <i>probability of not meeting the timing constraints after placement</i>. Clearly using cells with <i>sizes which are larger than necessary is wasteful in both silicon area and power consumption</i>. The <i>chips</i> thus synthesized will be <i>larger, cost more to produce and use more electrical power than necessary</i>.</p>	<p><u>'446 PATENT AT 1:61-2:3</u></p> <p>A common method for dealing with <i>inaccurate net load estimates</i> is by <i>estimating the net load at a considerably larger value than typically estimated</i>. Although this method increases the <i>probability of meeting timing constraints after placement</i>, it <i>causes the sizes of the gates to be considerably larger than necessary</i>. Gates which are <i>larger than the necessary size are wasteful in both silicon area and power consumption</i>. This leads to <i>chips</i> which are <i>larger, more expensive to produce, and use more electrical power than necessary</i>.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 3</u></p> <p>A second <i>problem with the conventional approach</i> is that the effect of synthesis decisions is hard to calculate. <i>Performing timing analysis during optimization is very time consuming, and accounts for most of the run time of conventional synthesis systems</i>.</p>	<p><u>'446 PATENT AT 2:4-9</u></p> <p>Another <i>problem with the conventional circuit design approach</i> concerns the timing analysis required <i>during optimization and during placement</i>. The <i>timing analysis performed throughout the conventional circuit design process is very time consuming, and accounts for most of the run time of a conventional circuit design system</i>.</p>

Synopsis Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 4</u></p> <p>In step 105 the placement program will <i>modify the net lengths. Depending on which location was chosen for each cell, the length of each net can be different. As the length differs, the capacitive load of the net changes</i>, and as a result, the delay of the cell driving the net changes. <i>Therefore the delays which were carefully optimized during the logic synthesis, are very different after placement</i>, and the <i>optimization</i> of the <i>network</i> is not very good.</p>	<p><u>'446 PATENT AT 2:12-19</u></p> <p><i>Depending on the location chosen for each gate, each net length may be modified. As each net length is modified, the capacitive load of the net will change. Therefore, the delays, which were carefully optimized during the logic design, are very different in value after cell placement</i>, thereby contributing to poor <i>network optimization</i>.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 4</u></p> <p><i>Much of the progress in the state of the art can be characterized as increased integration.</i> This is represented in figure 1 as various feedback paths, which repeat and alternate steps. The general direction has been towards programs which do structuring, mapping, sizing and placement simultaneously. <i>It has led to increasingly complex software systems which are slow and difficult to design and maintain.</i></p>	<p><u>'446 PATENT AT 2:20-23</u></p> <p>Additionally, <i>much of the progress in the state of the art for digital circuit design can be characterized as increased integration which has led to increasingly complex software systems which are slow, and difficult to design and maintain.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 4</u></p> <p><i>Iterating between placement and sizing</i> has been especially hard to execute because placement programs are not sold by the same design automation software vendors as <i>logic synthesis programs</i>. Also they are not run by the same users: the logic synthesis program is often run by the <i>designer, who also wrote the HDL specification</i>. The <i>placement program is often run by the silicon chip manufacturer, after the design is considered complete.</i></p>	<p><u>'446 PATENT AT 2:24-30</u></p> <p>A further disadvantage with conventional design approaches is in the difficulty of <i>iterating between placement and sizing</i>, since the <i>logic synthesis program</i> is often operated by the <i>logic designer who also wrote the HDL specification</i>, but the <i>placement program is often operated by the silicon chip manufacturer, after the design is complete.</i></p>

Synopsys Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 6</u></p> <p>The <i>present invention maintains timing closure</i> after it has been achieved by <i>adjusting the size of the cell during or after placement</i>. The <i>adjustments compensate for the fact that the placement algorithm can assign different net lengths to different nets and that these lengths are difficult to predict before placement</i>.</p>	<p><u>'446 PATENT AT 16:23-29</u></p> <p>According to the <i>present invention, timing closure is maintained</i> after placement occurs of cells 836. To <i>maintain timing closure</i>, the size of a particular gate may be <i>adjusted during or after placement</i>. This <i>adjustment compensates for the fact that placement algorithm may assign different net lengths to different nets, and that these different net lengths are difficult to predict prior to the placement step</i>.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 8</u></p> <p>Wherever possible, the same <i>reference numbers</i> will be used throughout the <i>drawings to refer to the same or like parts</i>.</p>	<p><u>'446 PATENT AT 4:59-63</u></p> <p><i>Referring in detail now to the drawings wherein similar parts or steps of the present invention are identified by like reference numerals</i>, there is seen in FIG. 1 a schematic diagram of a host computer system 100 which is capable of implementing the present invention.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 9</u></p> <p>The cells <i>can be combinational "gates"</i> 207, 208, 209, <i>whose function is represented as an expression in the Boolean algebra, using AND, OR and NOT operators</i>, or the cells <i>can be registers</i> 205, 206.</p>	<p><u>'446 PATENT AT 5:13-17</u></p> <p>The gates <i>can be combinational gates whose function is represented as Boolean expression</i> based on, for example, the <i>operators AND, OR and NOT</i>. The gates <i>can also be registers</i>.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 9</u></p> <p><i>Each cell (e.g., 208) has one or more inputs 212, 213, and a single output 214.</i></p>	<p><u>'446 PATENT AT 5:18-19</u></p> <p><i>Each gate (e.g., gate j) has one or more input 155 and a single output 160.</i></p>

Synopsys Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 9</u></p> <p><i>Cells whose inputs are connected to the output of a cell are called the fanin of the latter cell. Cells whose inputs are connected to the output of a cell are called the fanout of the latter cell.</i></p>	<p><u>'446 PATENT AT 5:26-32</u></p> <p><i>Gates whose outputs are connected to the inputs of a gate are collectively called the "fanin" of the latter gate. Thus, the gate k is in the fanin of the gate i. Gates whose inputs are connected to the output of a gate are collectively called the "fanout" of the latter gate. Thus, the gate j is in the fanout of the gate i.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 9</u></p> <p><i>The digital network performs a logic "function" by processing digital binary input data in a number of cycles. The input data is presented to the network on its *primary inputs* 201, 202, and the result of the computation of the network function is presented at the *primary outputs* 203, 204, of the network. The computation of the function takes one or more cycles. During each cycle the gate functions are calculated. The results are stored in the registers for use in the next cycle.</i></p>	<p><u>'446 PATENT AT 5:33-41</u></p> <p><i>The digital circuit 150 performs a logic function by processing digital binary input data in a number of cycles. The input data is presented to the digital circuit 150 at the primary inputs 170, and the result of the computation of the digital circuit function is presented at the primary outputs 175. Typically, the computation of the digital circuit function requires one or more cycles. During each cycle, the gate functions are calculated, and the calculation results are stored in registers for use in the next cycle.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 10</u></p> <p><i>The "arrival time" of the data at a gate is computed by taking the maximum arrival time of its fanin cells each increased by the delay from the input pin to the output pin.</i></p>	<p><u>'446 PATENT AT 9:55-58</u></p> <p><i>(An arrival time of the data at a gate is computed by taking the maximum arrival time of the fanin gates plus the delay measured from the input pin to the output pin of the gate).</i></p>

Synopsis Draft Patent Application	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 11</u></p> <p>The difference between the required time and the arrival time is the <i>*slack*</i>. If the arrival time is smaller than the required time, <i>the timing constraints are met</i>, and the <i>slack is positive</i>. If the arrival time is larger than the required time, the timing constraints are not met, and the slack is negative. The arrival time and required time may be different depending on whether the data is zero (0) or one (1). There also may be multiple arrival times and multiple required times to model a variety of timing constraints. <i>All slacks can be summarized as a single worst slack number</i>, called <i>the *network slack*</i>. <i>Timing closure is achieved if the network slack is non-negative</i>.</p>	<p><u>'446 PATENT AT 13:27-34</u></p> <p>This determination is made by subtracting the delay of the buffer from the "local <i>slack</i>", to give the value of the predicted slack after buffer insertion. <i>Slack is zero or positive if the timing constraints are met</i>. In addition, <i>all slacks in the circuit can be summarized by the "network slack" which is the single "worst" slack number</i>. <i>If the network slack is non-negative</i>, then the <i>timing closure is achieved</i>.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 11</u></p> <p>It is important to note that the dependency on size and load can be captured as the dependency on a single parameter C/S, and <i>the delay D is non-negative and monotonically increasing with C/S</i>.</p>	<p><u>'446 PATENT AT 6:38-43</u></p> <p>The delay D of a gate can be approximated by equation (1):</p> $D=f(C/S) \quad (1)$ <p><i>The delay D is non-negative and increases as the C/S value increases.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 11</u></p> <p><i>The delay may be different for different inputs of the gate and it may be different for the falling and the rising transition.</i></p>	<p><u>'446 PATENT AT 6:58-61</u></p> <p><i>The delay D value may also be different for different inputs of the gate and it may also be different for the falling transition and rising transition of a signal propagating through the gate.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 12</u></p> <p><i>The library analysis will determine a good value for C/S for each cell in the library.</i></p>	<p><u>'446 PATENT AT 6:63-65</u></p> <p><i>The library analysis will determine a "good" value for C/S for each gate in the library based on gain considerations.</i></p>

Synopsis Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 13</u></p> <p>Mostly these optimizations <i>change the structure of the network, and the Boolean functions of the cells, without changing the overall function of the network</i>. The types of <i>optimizations</i> that should be performed are <i>behavioral optimization such as resource sharing, sequential optimizations such as retiming, algebraic optimizations such as kernel extraction and Boolean optimizations such as redundancy removal</i>. There is a large amount of literature on how each of these <i>classes of optimizations</i> can be performed.</p>	<p><u>'446 PATENT AT 9:13-22</u></p> <p>During this step, <i>the structure of the circuit and the Boolean functions of the gates are changed</i> to reduce the total number of connections, <i>without changing the overall function of the circuit</i>. Structural <i>optimizations</i> can include <i>behavioral optimizations (such as resource sharing), sequential optimizations (such as retiming), algebraic optimizations (such as kernel extraction), and Boolean optimizations (such as redundancy removal)</i>. The <i>classes of optimizations</i> above are well known to those skilled in the art.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 13</u></p> <p>Following the library independent optimizations, the network <i>is mapped to a library of cells</i>. This means that <i>the logic functions of the cells are implemented with actual cells from the library</i>.</p>	<p><u>'446 PATENT AT 9:25-27</u></p> <p>In step 210 (FIG. 4), the circuit <i>is mapped to a library 209 of cells</i>. Thus, <i>the logic functions of the circuit gates are implemented with actual cells from the library 209</i>.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 14</u></p> <p>For example, we can <i>use the "boundary move" transformation to reduce the number of levels in the logic in the mapped network</i>. The boundary move <i>transform</i>, illustrated in fig x, <i>reduces the number of levels by bringing connection x forward</i>.</p>	<p><u>'446 PATENT AT 10:45-49</u></p> <p>A local <i>transformation</i> is then <i>used to reduce the number of levels in the logic in the gate chain circuit 550</i>. The result of the <i>transformation</i> is shown as gate chain circuit 550' in FIG. 7B. <i>The number of levels in the logic is reduced by bringing the gate 555 forward</i>.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 14</u></p> <p>To make the change legal <i>it is necessary that gates x, y and z are fanout free. If not, they must be made fanout free by making a copy</i>.</p>	<p><u>'446 PATENT AT 10:59-62</u></p> <p>In order for the transformation shown in FIG. 7B to be valid, <i>it is necessary that gates 555, 560, and 565 are fanout free. If the gates 555, 560, and 565 are not fanout free, then they are made fanout free through copying logic</i>.</p>

Synopsis Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 14-15</u></p> <p>In the <i>conventional</i> approach to logic synthesis, <i>copying logic will increase the load on gates x, x, x</i> and therefore increase the delay. <i>To predict if the transformation will improve delay, or hurt delay, it was necessary to run a complete static timing analysis with accurate delay models. If the change actually worsened the delay, then the change would be undone.</i></p>	<p><u>'446 PATENT AT 11:4-13</u></p> <p>Under <i>conventional</i> logic design, <i>copying logic will increase the load on the gates</i> whose outputs are connected to lines 575, 580, 585, and 590. In the example of FIG. 7B, the copying logic 555' <i>increases the load on the gates</i> whose outputs are connected to lines 575 and 580. <i>To predict whether or not the transformation improved delay, it is necessary to run a complete static timing analysis with accurate delay models. If the transformation (from circuit 550 to 550') were actually harmful to delay, then the transformation would have to be undone.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 15</u></p> <p><i>In the constant delay model approach, the effect of this change can be easily predicted. Note that changes in loads do not affect delay. The only change that affects delay, is the change of the fanin of gate xxx. The delay can easily be predicted by simple addition of gate delays.</i></p>	<p><u>'446 PATENT AT 10:49-58</u></p> <p><i>In the constant delay model approach, the effect of this transformation can be easily predicted. Changes in the gate loads do not affect delay, since delay is maintained as constant while gate size will be adjusted (during or after placement) to compensate for the load change. The only change which affects delay (of the gate chain circuit 550) is the change of the fanin of gate 555. This delay change can be predicted by simple addition of gate delays provided by the fanins connected at lines 590, 575, and 580 (see gate chain circuit 550' in FIG. 7B).</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 15</u></p> <p>The <i>net load</i> consists of the <i>load of the net</i>, which <i>can be estimated using a conventional net load model, plus any other fixed load, such as the load of a primary output.</i></p>	<p><u>'446 PATENT AT 11:26-30</u></p> <p>The parameter <i>w</i> represents the <i>net</i> (wire) <i>load</i> for a given gate <i>i</i> (wherein the <i>net load can be estimated using a conventional net load model</i> such as the above-mentioned fanout-based model) <i>plus any other fixed load such as the load of the primary output</i> of the circuit implementation.</p>

Synopsis Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 15</u></p> <p>In <i>a combinational network</i> this can be achieved by starting <i>at the primary outputs and traversing the network in a leveled order towards the primary inputs.</i></p>	<p><u>'446 PATENT AT 11:48-52</u></p> <p>If the digital circuit is <i>a combinational network</i> (see, e.g. circuit 150 in FIG. 2), then gate load calculation initiates <i>at the primary outputs 175 and traverses the circuit in a leveled order toward the primary inputs 170.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 15-16</u></p> <p>In <i>a sequential network there may be one or more loops, resulting a cyclic dependency: there is no rightmost cell.</i> In this case the computation <i>can start anywhere in the cycle, and repeats the cycle several times, until the capacitances converge</i> and the error is <i>sufficiently small.</i> It is possible that this iteration will <i>not converge and</i> that the <i>capacitance will increase</i> in every iteration, <i>by progressively larger amounts.</i> This situation is <i>detected</i> by requiring the increment to be smaller than <i>a preset maximum after a fixed number of iterations.</i> The iteration does <i>not converge</i> if the network is <i>an infeasible solution:</i> The current network cannot be <i>expected to work at this speed because its gain is too small. Changes</i> need to be made to the network to <i>increase the gain, which will usually mean increasing the delay</i> of the network as well.</p>	<p><u>'446 PATENT AT 11:53-12:4</u></p> <p>If the digital circuit is <i>a sequential network</i> (see, e.g., circuit 180 of FIG. 3), then <i>there may be one or more loops</i> (e.g., loop 182) which <i>result in a cyclic dependency</i> (i.e., <i>there is no "rightmost" gate</i>). Gate load calculation <i>can start anywhere in the cycle, and</i> calculation in <i>the cycle</i> is performed <i>several times until the load capacitance values converge</i> or have <i>sufficiently small</i> differences. However, a condition may exist when the load <i>capacitance</i> values do <i>not converge and increase by progressively larger amounts</i> every cycle calculation. This increase in load capacitance values can be <i>detected</i> if the calculated load values exceed <i>a preset maximum value after a fixed number of cycle calculations.</i> When the calculated load values do <i>not converge</i>, then the particular circuit 180 has <i>an infeasible solution</i>, which indicates that the digital circuit is <i>not expected to work at the set speed because the circuit gain is too small. Changes</i> are required to <i>increase the circuit gain</i>, and these changes <i>will usually</i> lead to an increase in circuit <i>delay.</i></p>

Synopsis Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 16</u></p> <p>After the loads have been calculated the <i>size</i> can be calculated <i>by dividing the actual load by the predetermined typical load</i>. The input capacitance can be calculated by multiplying the unit gate input capacitance by the size. The ratio of these numbers is the <i>size</i> of the gate. <i>The size is a scale factor, which</i> can be applied to the area of <i>the gate</i>, to give <i>the area of the sized gate</i>. <i>The area of the network can be estimated as the sum total of the areas of the sized gates, plus the net area as estimated from the total length of all nets.</i></p>	<p><u>'446 PATENT AT 12:5-20</u></p> <p>In the above example, the <i>size</i> S of a gate i is determined <i>by dividing the actual load C_i by the predetermined typical load C/S</i> of the gate i. The size S is a scale factor which is applied to all transistor channel widths of a gate in order to determine the area of the "<i>sized gate</i>". <i>The size S is also a scale factor which</i> is used to scale <i>the gate's</i> output load driving capability and its input pin loads. <i>The area of the sized gate</i> is determined by equation (5).</p> <p>area of sized gate=S*(area of gate) (5)</p> <p><i>The area of the mapped digital circuit can be estimated based on the sum of the total areas of the sized gates plus the net area</i> (which is estimated from the total length of all nets in the circuit).</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 16-17</u></p> <p>We can do this by <i>calculating</i> single parameter per net, called the <i>net weight</i>, which <i>represents the sensitivity of the total area of the network with respect to the load on that net</i>. This <i>net weight</i> can be calculated in a manner that is very similar to the calculation of the pin loads during the area calculation above. Starting at <i>the primary inputs, the net weight of the first (left-most) gate is equal to its area per unit load</i>. <i>The net weights of the other cells can now be calculated</i> with a recurrence relation traversing the network from left to right.</p>	<p><u>'446 PATENT AT 12:22-30</u></p> <p>Thus, the following discussion now turns to the <i>calculation of "net weights."</i> The <i>net weight represents the sensitivity of the total area of a digital circuit with respect to the load of a particular net</i>. As an example, <i>the net weight of a given gate, which is immediately coupled to the primary inputs of a digital circuit, is equal to its area per unit load</i>. Using equation (6), <i>the net weight of the other gates in the digital circuit are then calculated</i> in a leveled order towards the primary outputs of the digital circuit.</p>

Synopsis Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 17</u></p> <p>The <i>buffering</i> algorithm works as follows: First it finds <i>locations in the network where a buffer can be added</i> without increasing the network delay. This is done <i>by subtracting the delay of the buffer from the local slack, to give the predicted slack after buffer insertion. If the predicted slack is larger then the network slack, then a buffer can be inserted without increasing the network delay.</i></p>	<p><u>'446 PATENT-AT 13:23-37</u></p> <p>The <i>buffering</i> step of 215 (FIG. 4) is discussed in further detail with reference to FIG. 8. In step 650, <i>locations in the circuit are determined where a buffer can be added</i> so that buffer insertion will still permit timing constraints to be met. This determination is made <i>by subtracting the delay of the buffer from the "local slack", to give the value of the predicted slack after buffer insertion.</i> Slack is zero or positive if the timing constraints are met. In addition, all slacks in the circuit can be summarized by the "network slack" which is the single "worst" slack number. If the network slack is non-negative, then timing closure is achieved. <i>If the predicted slack calculated in step 650 is larger than the network slack, then it is possible to insert a buffer without increasing the circuit delay.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 17-18</u></p> <p>Next we have to calculate the reduction in load of this net, and check that <i>area that is added by adding the buffer does not exceed the area saved by sizing down the source gate.</i> The <i>area added by inserting the buffer</i> is simply <i>the area of the buffer times its size, where the size is determined by the load on the buffer divided by the typical load of the buffer.</i> The <i>area saved by inserting the buffer</i> can be calculated by first <i>calculating the change in load due to the insertion of the buffer: some sinks are removed, the input load of the buffer is added, and the net load estimate may change as a result of the number of fanouts of the net changing.</i></p>	<p><u>'446 PATENT AT 13:37-48</u></p> <p>In step 655, it is determined whether the <i>added area</i> due to <i>buffer</i> insertion <i>does not exceed the area saved by sizing down the source gate.</i> The <i>added area (by inserting the buffer)</i> is equal to <i>the area of the buffer multiplied by the buffer size, wherein the buffer size is determined by the buffer load C divided by the typical load C/S on the buffer.</i> The <i>area saved by sizing down the source gate</i> is determined by first <i>calculating the change in net load due to the buffer insertion.</i> This <i>net load change</i> is due to the following: (1) <i>some sinks (which sink currents) are removed,</i> (2) <i>the input load of the buffer is added, and</i> (3) <i>the number of fanouts of the gate may change.</i></p>

Synopsys Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 18</u></p> <p><i>After the buffer is inserted, the capacitances need to be updated in the fanin cone of the buffer, while the net weights need to be updated in the fanout cone of the buffer.</i></p>	<p><u>'446 PATENT AT 13:53-57</u></p> <p><i>After the buffer has been inserted, then in step 670 the capacitance values need to be updated in the fanin cone of the buffer, while the net weights need to be updated in the fanout cone of the buffer.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 18</u></p> <p>The next step is the process of "Stretching" and "Compressing" the delays of the individual gates to meet the timing constraints. Gates which are on long paths which do not meet the delay constraint are "compressed" until the path does meet the timing constraint. Gates on the short paths which easily meet the timing constraints are "Stretched". Gates with stretched delays require less area for the same load. In this step the delay of the gates is traded against the gain of the gates. When the delay of a gate decreases so does the gain of the gate. It is important that there is enough gain in the network.</p>	<p><u>'446 PATENT AT 14:20-36</u></p> <p>Prior to cell placement, the delays of the individual gates may be stretched or compressed to meet the delay constraints, as shown in step 220 of FIG. 4. As shown in FIG. 9A, by compressing (decreasing) the delay of a given gate, the gate gain decreases. Gates which are on long paths not meeting the delay constraints are compressed (in delay) until the long paths meet the delay constraints. The delay of the gates (or gate) may be decreased as long as the minimum required gain requirements are met. By stretching (increasing) the delay of a given gate, the gate gain increases (see FIG. 9A). Gates on short paths which easily meet the delay constraints are stretched (in delay), since gates with stretched delays require less area for the same load. The delay of the gates (or gate) in a path are stretched to the extent that timing constraints for the digital circuit are still met.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 18-19</u></p> <p><i>For the purpose of stretching and compressing registers can usually be considered to be part of a path which they originate, but not of a path that they terminate.</i></p>	<p><u>'446 PATENT AT 15:48-51</u></p> <p><i>For the purpose of stretching and compressing, registers in the circuit are preferably considered as part of a path from which they originate, but not part of the path from which they terminate.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 19</u></p> <p>The stretching algorithm considers the cells on a path by path basis, processing the path with the smallest slack first.</p>	<p><u>'446 PATENT AT 15:9-10</u></p> <p>The invention operates on a path-by-path basis whereby the most critical path in a digital circuit 750 is evaluated first.</p>

Synopsis Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 19</u></p> <p>After a cell <i>has been adjusted</i>, it becomes "<i>locked</i>" and its <i>delay</i> cannot be changed by the <i>stretching</i> algorithm.</p>	<p><u>'446 PATENT AT 15:21-25</u></p> <p>After the gate 754 <i>has been adjusted</i> to meet the Path 2 timing constraints, <i>it becomes "locked,"</i> whereby the gate 754 delay will <i>not be</i> adjusted further for the remainder of the compression and <i>stretching</i> step.</p>

1 Chris Scott Graham (State Bar No. 114498)
2 Michael N. Edelman (State Bar No. 180948)

3 **DECHERT LLP**
4 975 Page Mill Road
5 Palo Alto, California 94304
6 Telephone: (650) 813-4800
7 Facsimile: (650) 813-4848

8 Attorneys for Plaintiff SYNOPSISYS

ORIGINAL
FILED

SEP 17 2004

RICHARD W. WIEKING
CLERK, U.S. DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
SAN JOSE

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN JOSE DIVISION

12 SYNOPSISYS, INC., a Delaware corporation,

13 Plaintiff,

14 vs.

15 MAGMA DESIGN AUTOMATION, a
16 Delaware corporation,

17 Defendant.

CASE NO. **C04 03923 MEJ**
COMPLAINT FOR PATENT
INFRINGEMENT
ADR
DEMAND FOR JURY TRIAL
E-filing

18
19 Plaintiff SYNOPSISYS, INC. ("SYNOPSISYS") hereby alleges against Defendant MAGMA
20 DESIGN AUTOMATION ("MAGMA" or "the Defendant") as follows:

21 **JURISDICTION**

22 1. This is an action for patent infringement arising under the patent laws of the United
23 States. This Court has jurisdiction over this action under 28 U.S.C. § 1338(a).

24 **PARTIES**

25 2. SYNOPSISYS is a corporation duly organized and existing under the laws of the State
26 of Delaware, with its principal place of business in Mountain View, California.

27 ///

28 ///

3. SYNOPSISYS is informed and believes, and thereon alleges, that MAGMA is a corporation duly organized and existing under the laws of the State of Delaware, with its principal place of business in Santa Clara, California.

VENUE

4. Venue is proper in the Northern District pursuant to 28 U.S.C. § 1391(b) & (c) and 28 U.S.C. § 1400(b).

INTRADISTRICT ASSIGNMENT

5. This is an Intellectual Property Action under this Court's Assignment Plan, and therefore assignment to any division of the Court is proper pursuant to Civil L.R. 3-2(c). SYNOPSYS believes that assignment to the San Jose division is particularly appropriate given its close proximity to the principal places of business of SYNOPSYS and MAGMA.

FACTUAL BACKGROUND

6. On or about May 17, 1995, Lukas van Ginneken signed a Proprietary Information and Inventions Agreement (the "Agreement") as a condition to his employment by SYNOPSISYS. Paragraph 3 of this Agreement provides that all rights to any inventions made, conceived, reduced to practice or developed by van Ginneken while employed by SYNOPSISYS are automatically assigned to SYNOPSISYS. A true and correct copy of the Agreement is attached hereto as Exhibit A.

7. While employed by SYNOPSYS, van Ginneken made, conceived and developed inventions pertaining to timing closure methodology, the use of constant delay models in logic synthesis and other aspects of placement and/or synthesis. These inventions were made, conceived and developed by van Ginneken during his employment for SYNOPSYS for the purpose of developing SYNOPSYS' products, and therefore each of these inventions are encompassed by the terms of the Agreement. By operation of law, all right, title and interest to these inventions are automatically assigned to SYNOPSYS under the Agreement.

8. After leaving the employment of SYNOPSIS, van Ginneken co-founded MAGMA. Thereafter, MAGMA submitted patent applications to the Patent and Trademark Office that disclosed inventions that van Ginneken had made, conceived and developed while at SYNOPSIS, and which are owned by SYNOPSIS.

9. On April 23, 2002, United States Patent No. 6,378,114 ("the '114 Patent"), entitled "Method for the Physical Placement of an Integrated Circuit Adaptive to Netlist Changes," was issued to SYNOPSYS. van Ginneken is a named inventor on the '114 Patent. A true and correct copy of the '114 Patent is attached to this complaint as Exhibit B and is incorporated by reference herein.

10. On September 17, 2002, United States Patent No. 6,453,446 ("the '446 Patent"), entitled "Timing Closure Methodology," was issued to MAGMA. The '446 Patent discloses inventions which were made, conceived and developed by van Ginneken at SYNOPSYS. Pursuant to the terms of the Agreement, SYNOPSYS holds legal and equitable title to the inventions in the '446 Patent. A true and correct copy of the '446 Patent is attached to this complaint as Exhibit C and is incorporated by reference herein.

11. On April 20, 2004, United States Patent No. 6,725,438 ("the '438 Patent"), entitled "Timing Closure Methodology," was issued to MAGMA. The '438 Patent contains inventions which were made, conceived and developed by van Ginneken at SYNOPSYS. Pursuant to the terms of the Agreement, SYNOPSYS holds legal and equitable title to the inventions in the '438 Patent. A true and correct copy of the '438 Patent is attached to this complaint as Exhibit D and is incorporated by reference herein.

12. Since the issuance of the '114 Patent, '446 Patent, and '438 Patent (collectively referred to hereinafter as the "SYNOPSISYS PATENTS"), MAGMA has engaged in a wide range of activities to infringe those patents. MAGMA has been involved in making, using, selling, distributing, advertising, marketing and creating source code for products that infringe the SYNOPSISYS PATENTS.

FIRST CAUSE OF ACTION
(PATENT INFRINGEMENT)

13. SYNOPSISYS is the owner of the SYNOPSISYS PATENTS because, among other reasons, the inventions disclosed in the patents were previously assigned to SYNOPSISYS by van Ginneken pursuant to the terms of the Agreement.

111

1 14. MAGMA has been and still is infringing the SYNOPSIS PATENTS in violation of
2 the federal patent laws by making, using, selling, distributing, advertising, marketing and creating
3 source code for products which infringe the SYNOPSIS PATENTS. MAGMA will continue to so
4 infringe unless enjoined by this Court.

5 15. MAGMA has actively induced infringement of, or contributed to the infringement of,
6 the SYNOPSIS PATENTS under the federal patent laws by, among other things, making infringing
7 products and creating source code for infringing products and then selling, distributing, advertising
8 and marketing those infringing products to others, and will continue to do so unless enjoined by this
9 Court.

10 16. MAGMA's infringement of the SYNOPSIS PATENTS in violation of the federal
11 patent laws has been willful and deliberate, and has caused injury to SYNOPSIS.

12 17. MAGMA's infringement in violation of the federal patent laws will continue to injure
13 SYNOPSIS unless enjoined by this Court.

14 WHEREFORE, SYNOPSIS prays for judgment against the Defendant, and requests that this
15 Court impose the following remedies under the federal patent laws:

16 A. Preliminarily and permanently enjoin the Defendant from continued infringement of
17 the SYNOPSIS PATENTS, pursuant to 35 U.S.C. § 283;

18 B. Order the Defendant to account to SYNOPSIS for damages sustained by
19 SYNOPSIS as a result of the Defendant's infringement of the SYNOPSIS PATENTS, with
20 interest, pursuant to 35 U.S.C. § 284;

21 C. Order the Defendant to pay SYNOPSIS a reasonable royalty to compensate for the
22 Defendant's infringement, pursuant to 35 U.S.C. § 284;

23 D. Treble the damages resulting from the Defendant's willful and deliberate
24 infringement, pursuant to 35 U.S.C. § 284;

25 E. Award SYNOPSIS its costs, expenses and reasonable attorneys' fees incurred in
26 bringing and prosecuting this action, pursuant to 35 U.S.C. § 285;

27 ///

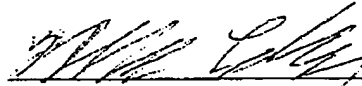
28 ///

1 F. Impose a constructive trust for the benefit of SYNOPSIS over any profits, revenues,
2 or other benefits obtained by the Defendant as a result of its infringement of the SYNOPSIS
3 PATENTS; and

4 G. Award SYNOPSIS such further relief that the Court may deem just and proper
5 arising from the Defendant's infringement of the SYNOPSIS PATENTS under the federal patent
6 laws.

7 Dated: September 17, 2004

DECHERT LLP

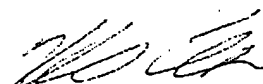
8 
9 Chris Scott Graham
10 Michael Edelman
11 Attorneys for Plaintiff SYNOPSIS
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

DEMAND FOR JURY TRIAL

SYNOPSISYS hereby demands trial by jury of all issues.

Dated: September 17, 2004

DECHERT LLP



Chris Scott Graham

Michael Edelman

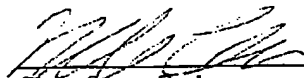
Attorneys for Plaintiff SYNOPSISYS

1 CERTIFICATION OF INTERESTED ENTITIES OR PERSONS

2 Pursuant to Civil L.R. 3-16, the undersigned certifies that as of this date, other than the
3 named parties, there is no such interest to report.

4 Dated: September 17, 2004

DECHERT LLP

5 
6 Chris Scott Graham
7 Michael Edelman
8 Attorneys for Plaintiff SYNOPSYS

PROPRIETARY INFORMATION
AND
INVENTIONS AGREEMENT

The following confirms an agreement between me and Synopsys, Inc. (the "Company"), which is a material part of the consideration for my employment by the Company.

1. I recognize that the Company is engaged in a continuous program of research, development and production respecting its business, present and future, including fields generally related to its business and that the Company possesses and continues to possess information that has been created, discovered, developed or otherwise become known to the Company (including, without limitation, information created, discovered or developed by, or made known to, me during the period of or arising out of my employment by the Company) and/or in which property rights have been assigned, licensed or otherwise conveyed to the Company, which information has commercial value in the business in which the Company is engaged. All of the aforementioned information is hereinafter called "Proprietary Information." By way of illustration, but not limitation, Proprietary Information includes trade secrets, processes, data and know-how, computer software, improvements, inventions, works of authorship, techniques, marketing plans, strategies, forecasts and copyrightable material and customer lists.

2. I understand that my employment creates a relationship of confidence and trust between me and the Company with respect to any information:

(i) applicable to the business of the Company; or

(ii) applicable to the business of any client or customer of the Company, which may be known to me by the Company or by any client or customer of the Company, or learned by me during the period of my employment.

3. In consideration of my employment by the Company and the compensation received by me from the Company from time to time, I hereby agree as follows:

A. All Proprietary Information shall be the sole property of the Company and its assigns, and the Company and its assigns shall be sole owner of all patents, copyrights and other rights in connection therewith. I hereby assign to the Company any rights I may have or acquire in such Proprietary Information. At all times, both during my employment by the Company and after its termination, I will keep in confidence and trust all Proprietary Information, and I will not use or disclose any Proprietary Information or anything relating to it without the written consent of the Company, except as may be necessary in the ordinary course of performing my duties to the Company.

B. All documents, records, apparatus, equipment and other physical property, whether or not pertaining to Proprietary Information, furnished to me by the Company or produced by me or others in connection with my employment shall be and remain the sole property of the Company and shall be returned to the Company immediately as and when requested by the Company. Even if the Company does not so request, I shall return and deliver all such property upon termination of my employment by me or the Company for any reason and I will not take with me any such property or any reproduction of such property upon such termination.

C. I will promptly disclose to the Company, or any persons designated by it, all improvements, inventions, works of authorship, processes, techniques, know-how, formulae,

data, ideas and other information (including, without limitation, my algorithms or software), whether or not patentable, made or conceived or reduced to practice or learned by me, either alone or jointly with others, during the term of my employment (all said improvements, inventions, works of authorship, processes, techniques, know-how, formulae, data, ideas and other information shall be hereinafter collectively called "Inventions").

D. I agree that all Inventions which I make, conceive, reduce to practice or develop (in whole or in part, either alone or jointly with others) during my employment shall be the sole property of the Company to the maximum extent permitted by Section 2870 of the California Labor Code (hereinafter called "Section 2870"), a copy of which is attached hereto as Exhibit B, and to the extent permitted by law, shall be "works made for hire." The Company shall be the sole owner of all patents, copyrights, trade secret rights, rights with respect to other intellectual property or other rights in connection therewith (including, without limitation, such rights in algorithms or software). I hereby assign to the Company any rights I may have or acquire in such Inventions. I agree to perform, during and after my employment, all acts deemed necessary or desirable by the Company to permit and assist it, at the Company's expense, in obtaining, maintaining and enforcing patents, copyrights, trade secret rights, rights with respect to such Inventions and/or other Inventions I have or may at any time assign to the Company in any and all countries. Such acts may include, but are not limited to, execution of documents and assistance or cooperation in legal proceedings. I hereby irrevocably designate and appoint the Company and its duly authorized officers and agents as my agents and attorneys-in-fact to act for and on my behalf and instead of me, to execute and file any applications or related filings and to do all other lawfully permitted acts to further the prosecution, maintenance and enforcement, issuance of patents, copyrights, trade secret rights, rights with respect to mask works or other rights thereon with the same legal force and effect as if executed by me.

E. As a matter of record, I attach hereto a complete list of all inventions or improvements relevant to the subject matter of my employment by the Company which have been made by me or jointly with others prior to my employment with the Company that I desire to remove from the operation of this Agreement, and I covenant that such list is complete. If no such list is attached to this Agreement, I represent that I have no such inventions and improvements at the time of signing this Agreement.

F. During the term of my employment and for one (1) year thereafter, I will not encourage or solicit any employee of the Company to leave the Company for any reason or devote less than all of any such employee's efforts to the affairs of the Company, provided that the foregoing shall not affect any responsibility I may have as an employee if the Company with respect to the bona fide hiring and firing of Company personnel.

G. I represent that my performance of all the terms of this Agreement will not breach any agreement to keep in confidence proprietary information acquired by me in confidence or in trust prior to my employment by the Company. I have not entered into, and I agree I will not enter into, any agreement, either written or oral, in conflict herewith.

H. I represent that execution of this Agreement, my employment with the Company and my performance of my proposed duties to the Company in the development of its business will not violate any obligations I may have to my former employer.

I. This Agreement does not require assignment of an invention which an employee cannot be obligated to assign under Section 2870. However, I will disclose any Inventions as required by Section 3(c) hereof regardless of whether I believe the Invention is protected by Section 2870, in order to permit the Company to engage in a review process to determine such issues as may arise. Such disclosure shall be received in confidence by the Company.

4. This Agreement shall be effective as of the first day of my employment by the Company: June 26 1995 (date).

5. This Agreement shall be binding upon me, my heirs, executors, assigns and administrators and shall inure to the benefit of the Company, its successors and assigns.

Dated: 5/17/95


Signature

Lukas van Ginneken
(Print or Type Name)

Accepted and agreed to:

Synopsys, Inc.

By  _____

EXHIBIT A

TO

SYNOPSYS, INC. PROPRIETARY INFORMATION
AND INVENTIONS AGREEMENT

1. The following is a complete list of all inventions or improvements relevant to the subject matter of my employment by Synopsys, Inc. (the "Company") that have been made or conceived or first reduced to practice by me alone or jointly with others prior to my employment by the Company that I desire to remove from the operation of the Company's Proprietary Information and Inventions Agreement.

☐ No inventions or improvements

☐ See below

☒

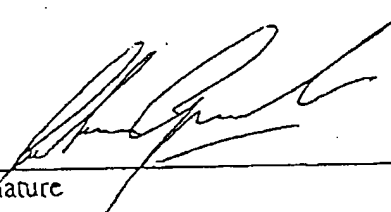
Additional sheets attached

2. I propose to bring to my employment the following materials and documents of a former employer:

☒

No materials or documents

☐ See below


Signature

Lukas van Gyncken
Print or Type Name

EXHIBIT B

TO

SYNOPSYS, INC. PROPRIETARY INFORMATION
AND INVENTIONS AGREEMENT

Section 2870. Application of provision providing that employee shall assign or offer to assign rights in inventions to employer.

(a) Any provision in an employment agreement which provides that an employee shall assign, or offer to assign, any of his or her rights in an invention to her or her employer shall not apply to an invention that the employee entirely on his or her own time without using the employer's equipment, supplies, facilities or trade secret information except for those inventions that either:

(1) Relate at the time of conception or reduction to practice of the invention to the employer's business, or actual or demonstrably anticipated research or development by the employer;

(2) result from any work performed by the employee for the employer.

(b) To the extent a provision in an employment agreement purports to require an employee to assign an invention otherwise excluded from being required to be assigned under subdivision (a), the provision is against the public policy of this state and is unenforceable.

The following is a complete list of all inventions or improvements relevant to the subject matter of my employment by Synopsys, Inc. (the Company) that have been made or conceived or first reduced to practice by me alone or jointly with others prior to my employment by the Company that I desire to remove from the operation of the Company's Proprietary Information and Inventions Agreement.

- [0] "Efficient orthonormality testing for synthesis with pass transistor selectors" by M. R. C. M. Berkelaar and -, accepted at the International Workshop on Logic Synthesis, June 1995.
- [1] "Timing Verification and Optimization for the PowerPC Processor Family", by R.E. Mains, T. A. Mosher, - and R.F. Damiano, in: Proc. Int. Conf. on Computer Design, pp.390-393, Boston, Oct. 10-12, 1994.
- [2] "In the driver's seat of BooleDozer" by D. Brand and R.F. Damiano, -, A. D. Drumm, in: Proc. Int. Conf. on Computer Design, pp. 518-521 Boston, Oct. 10-12, 1994.
- [3] "Grammar-based optimization of synthesis scenarios" by A. Kuehlmann and -, in: Proc. Int. Conf. on Computer Design, pp. 20-25 Boston, Oct. 10-12, 1994.
- [4] "Tuning of logic synthesis scenarios" by - and A. Kuehlmann, Workshop notes of the Int. Workshop on logic synthesis, paper P7c, Tahoe City, May 23-26, 1993.
- [5] "Fanin ordering in multi-slot timing" by -, Proc. Int. Conf. on Computer Design, pp. 44-47, Cambridge, Oct. 11-14, 1992.
- [6] "The complexity of adaptive annealing" by R. H. J. M. Otten and -, Proc. Int. Conf. on Computer Design, pp. 404-407, Cambridge, Sept. 17-19, 1990.
- [7] "Buffer placement in distributed RC-tree networks for minimal Elmore delay" by -, Proc. Int. Symp. on Circuits and Systems, pp. 865-868, New Orleans, May 2-5, 1990.
- [8] "Optimal slicing of plane point placements" by - and R. H. J. M. Otten, Proc. European Design Automation Conf. pp. 322-236, Glasgow, March 12-15, 1990.
- [9] The annealing algorithm by R. H. J. M. Otten and -, ISBN 07923-9022-9, Boston:Kluwer, 1989.
- [10] The predictor-adaptor paradigm - automation of custom layout by flexible design by -, Ph.D. thesis, ISBN 90-9002703-3, Eindhoven, 1989.
- [11] "Doubly folded transistor matrix layout" by - and J. T. J. van Eijndhoven, A. H. C. M. Brouwers, Digest Int. Conf. on Computer Aided Design, Santa Clara, Nov. 7-10, 1988.
- [12] "Stop criteria in simulated annealing" R. H. J. M. Otten and -, Proc.

Int. Conf. on Computer Design, pp.549-552, Port Chester, Oct. 3-5,
1988.

[13] "An inner loop criterion for simulated annealing" by - and R.H.J.M.
Otten, Physics letters A, 130:429-435, 1988.

[14] "Soft Macro Cell generation by two dimensional folding" by - and J.
T. J. van Eijndhoven, P. R. M. van Teeffelen, T. J. Deckers, Proc. Int.
Symp. on Circuits and Systems, pp. 727-730, Espoo, June 1988.

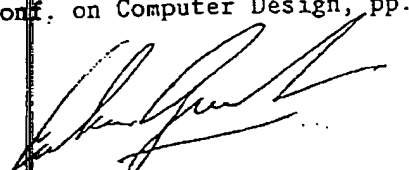
[15] "Gridless routing of general floor plans" by - and J. A. G. Jess, Digest
Int. Conf. on Computer Aided Design, pp. 30-33, Santa Clara Nov. 9-
12, 1987.

[16] "Wire planning for stackable designs", by R. K. Brayton, C. L. Chen,
J. A. G. Jess, R. H. J. M. Otten and -, Proc. Int. Symp. on VLSI tech-
nology, pp.269-273, Taipeh, May 13-15, 1987.

[17] "Global wiring for custom layout design" by - and R. H. J. M. Otten,
Proc. Int. Symp. on Circuits and Systems. pp.207-208, Kyoto, June 5-
7, 1985.

[18] "Floor plan design using simulated annealing" by R. H. J. M. Otten
and -, Digest Int. Conf. on Computer Aided Design, pp. 96-98, Santa
Clara, Nov. 1984.

[19] "Stepwise layout refinement" by - and R. H. J. M. Otten, Proc. Int.
Conf. on Computer Design, pp. 30-36, Port Chester, Oct.8-11, 1984.



Lukas van Ginneken

EV 31464 4795 28 JANUARY 2005

PTO/SB/82 (09-04)

Approved for use through 11/30/2005. OMB 0651-0035

U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

to respond to a collection of information unless it displays a valid OMB control number.

Under the

EV314044795US

**REVOCATION OF POWER OF
ATTORNEY WITH
NEW POWER OF ATTORNEY
AND
CHANGE OF CORRESPONDENCE ADDRESS**

Application Number	10/134,076 (Pat. 6,725,438)
Filing Date	24 April 2002 (Issued 20 Apr 2004)
First Named Inventor	van Ginneken
Art Unit	unknown
Examiner Name	Vuthe Siek
Attorney Docket Number	SYNP 103

I hereby revoke all previous powers of attorney given in the above-identified application.

☐ A Power of Attorney is submitted herewith.

OR

☒ I hereby appoint the practitioners associated with the Customer Number:

36454

☒ Please change the correspondence address for the above-identified application to:

☒ The address associated with
Customer Number:

36454

OR

☐ Firm or
Individual Name

Address

City

State

Zip

Country

Telephone

Fax

I am the:

☐ Applicant/Inventor.

☒ Assignee of record of the entire interest. See 37 CFR 3.71.
Statement under 37 CFR 3.73(b) is enclosed. (Form PTO/SB/96)

SIGNATURE of Applicant or Assignee of Record

Signature

Name

Date

Deirdre Hanford

28 January 2005

Telephone

(650) 584-4201

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below.

☐ *Total of _____ forms are submitted.

This collection of information is required by 37 CFR 1.36. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

STATEMENT UNDER 37 CFR 3.73(b)

Applicant/Patent Owner: van Ginneken

Application No./Patent No.: 09/054,379/6,453,446 Filed/Issue Date: 2 April 1998/17 September 2002

Entitled: TIMING CLOSURE METHODOLOGY

SYNOPSYS, INC., a CORPORATION
(Name of Assignee) (Type of Assignee, e.g., corporation, partnership, university, government agency, etc.)

states that it is:

1. ☒ the assignee of the entire right, title, and interest; or
2. ☐ an assignee of less than the entire right, title and interest.
The extent (by percentage) of its ownership interest is _____ %

in the patent application/patent identified above by virtue of either:

A. ☐ An assignment from the inventor(s) of the patent application/patent identified above. The assignment was recorded in the United States Patent and Trademark Office at Reel _____, Frame _____, or for which a copy thereof is attached.

OR

B. ☒ A chain of title from the inventor(s), of the patent application/patent identified above, to the current assignee as shown below:

1. From: Lukas P.P.P. van Ginneken To: Synopsys, Inc.
The document was recorded in the United States Patent and Trademark Office at
Reel _____, Frame _____, or for which a copy thereof is attached.
2. From: _____ To: _____
The document was recorded in the United States Patent and Trademark Office at
Reel _____, Frame _____, or for which a copy thereof is attached.
3. From: _____ To: _____
The document was recorded in the United States Patent and Trademark Office at
Reel _____, Frame _____, or for which a copy thereof is attached.

☒ Additional documents in the chain of title are listed on a supplemental sheet.

☒ Copies of assignments or other documents in the chain of title are attached.

[NOTE: A separate copy (i.e., a true copy of the original assignment document(s)) must be submitted to Assignment Division in accordance with 37 CFR Part 3, if the assignment is to be recorded in the records of the USPTO. See MPEP 302.08]

The undersigned (whose title is supplied below) is authorized to act on behalf of the assignee.

Deirdre Hanford
Signature

28 January 2005
Date

Deirdre Hanford
Printed or Typed Name

(650) 584-4201
Telephone Number

Sr. Vice President
Title

This collection of information is required by 37 CFR 3.73(b). The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Application No. 09/054,379
Patent 6,453,446

SUPPLEMENTAL SHEET TO STATEMENT UNDER 37 CFR 3.73(b)

Attached in support of the chain of title of the referenced application are copies of the following documents:

- 1) a copy of a Proprietary Information and Inventions Agreement attached hereto, signed by the inventor, that assigns to Synopsys, Inc. all inventions made, conceived, reduced to practice, or developed during the inventor's employment with Synopsys, Inc.;
- 2) a copy of a declaration by Robert Damiano, attached hereto, in which Robert Damiano attests that he received a draft patent application from the inventor ("Draft") during the inventor's employment with Synopsys, Inc., including as an attachment the email in which Robert Damiano received the Draft;
- 3) a copy of a chart, attached hereto, highlighting the common language shared between the Draft and the specification of U.S. Patent No. 6,453,446, from which U.S. Patent No. 6,725,438 claims priority as a continuation, from which U.S. Patent Application No. 10/828,547 claims priority as a continuing application, including as an attachment the Draft with page numbers referenced by the chart; and
- 4) a copy of a Complaint for Patent Infringement attached hereto for purposes of disclosure.

PROPRIETARY INFORMATION
AND
INVENTIONS AGREEMENT

The following confirms an agreement between me and Synopsys, Inc. (the "Company"), which is a material part of the consideration for my employment by the Company.

1. I recognize that the Company is engaged in a continuous program of research, development and production respecting its business, present and future, including fields generally related to its business and that the Company possesses and continues to possess information that has been created, discovered, developed or otherwise become known to the Company (including, without limitation, information created, discovered or developed by, or made known to, me during the period of or arising out of my employment by the Company) and/or in which property rights have been assigned, licensed or otherwise conveyed to the Company, which information has commercial value in the business in which the Company is engaged. All of the aforementioned information is hereinafter called "Proprietary Information." By way of illustration, but not limitation, Proprietary Information includes trade secrets, processes, data and know-how, computer software, improvements, inventions, works of authorship, techniques, marketing plans, strategies, forecasts and copyrightable material and customer lists.

2. I understand that my employment creates a relationship of confidence and trust between me and the Company with respect to any information:

(i) applicable to the business of the Company; or

(ii) applicable to the business of any client or customer of the Company, which may be known to me by the Company or by any client or customer of the Company, or learned by me during the period of my employment.

3. In consideration of my employment by the Company and the compensation received by me from the Company from time to time, I hereby agree as follows:

A. All Proprietary Information shall be the sole property of the Company and its assigns, and the Company and its assigns shall be sole owner of all patents, copyrights and other rights in connection therewith. I hereby assign to the Company any rights I may have or acquire in such Proprietary Information. At all times, both during my employment by the Company and after its termination, I will keep in confidence and trust all Proprietary Information, and I will not use or disclose any Proprietary Information or anything relating to it without the written consent of the Company, except as may be necessary in the ordinary course of performing my duties to the Company.

B. All documents, records, apparatus, equipment and other physical property, whether or not pertaining to Proprietary Information, furnished to me by the Company or produced by me or others in connection with my employment shall be and remain the sole property of the Company and shall be returned to the Company immediately as and when requested by the Company. Even if the Company does not so request, I shall return and deliver all such property upon termination of my employment by me or the Company for any reason and I will not take with me any such property or any reproduction of such property upon such termination.

C. I will promptly disclose to the Company, or any persons designated by it, all improvements, inventions, works of authorship, processes, techniques, know-how, formulae,

ATTORNEYS AND
CONSULTANTS ONLY

SY000003

data, ideas and other information (including, without limitation, my algorithms or software), whether or not patentable, made or conceived or reduced to practice or learned by me, either alone or jointly with others, during the term of my employment (all said improvements, inventions, works of authorship, processes, techniques, know-how, formulae, data, ideas and other information shall be hereinafter collectively called "Inventions").

D. I agree that all Inventions which I make, conceive, reduce to practice or develop (in whole or in part, either alone or jointly with others) during my employment shall be the sole property of the Company to the maximum extent permitted by Section 2870 of the California Labor Code (hereinafter called "Section 2870"), a copy of which is attached hereto as Exhibit B, and to the extent permitted by law, shall be "works made for hire." The Company shall be the sole owner of all patents, copyrights, trade secret rights, rights with respect to other intellectual property or other rights in connection therewith (including, without limitation, such rights in algorithms or software). I hereby assign to the Company any rights I may have or acquire in such Inventions. I agree to perform, during and after my employment, all acts deemed necessary or desirable by the Company to permit and assist it, at the Company's expense, in obtaining, maintaining and enforcing patents, copyrights, trade secret rights, rights with respect to such Inventions and/or other Inventions I have or may at any time assign to the Company in any and all countries. Such acts may include, but are not limited to, execution of documents and assistance or cooperation in legal proceedings. I hereby irrevocably designate and appoint the Company and its duly authorized officers and agents as my agents and attorneys-in-fact to act for and on my behalf and instead of me, to execute and file any applications or related filings and to do all other lawfully permitted acts to further the prosecution, maintenance and enforcement, issuance of patents, copyrights, trade secret rights, rights with respect to mask works or other rights thereon with the same legal force and effect as if executed by me.

E. As a matter of record, I attach hereto a complete list of all inventions or improvements relevant to the subject matter of my employment by the Company which have been made by me or jointly with others prior to my employment with the Company that I desire to remove from the operation of this Agreement, and I covenant that such list is complete. If no such list is attached to this Agreement, I represent that I have no such inventions and improvements at the time of signing this Agreement.

F. During the term of my employment and for one (1) year thereafter, I will not encourage or solicit any employee of the Company to leave the Company for any reason or devote less than all of any such employee's efforts to the affairs of the Company, provided that the foregoing shall not affect any responsibility I may have as an employee if the Company with respect to the bona fide hiring and firing of Company personnel.

G. I represent that my performance of all the terms of this Agreement will not breach any agreement to keep in confidence proprietary information acquired by me in confidence or in trust prior to my employment by the Company. I have not entered into, and I agree I will not enter into, any agreement, either written or oral, in conflict herewith.

H. I represent that execution of this Agreement, my employment with the Company and my performance of my proposed duties to the Company in the development of its business will not violate any obligations I may have to my former employer.

I. This Agreement does not require assignment of an invention which an employee cannot be obligated to assign under Section 2870. However, I will disclose any Inventions as required by Section 3(c) hereof regardless of whether I believe the Invention is protected by Section 2870, in order to permit the Company to engage in a review process to determine such issues as may arise. Such disclosure shall be received in confidence by the Company.

SY000004

**ATTORNEYS AND
CONSULTANTS ONLY**

4. This Agreement shall be effective as of the first day of my employment by the Company: June 26 1995 (date).

5. This Agreement shall be binding upon me, my heirs, executors, assigns and administrators and shall inure to the benefit of the Company, its successors and assigns.

Dated: 5/17/95




Signature

Lukas van Ginneken

(Print or Type Name)

Accepted and agreed to:

Synopsys, Inc.

By  _____

ATTORNEYS AND
CONSULTANTS ONLY

SY000005

EXHIBIT A
TO
SYNOPSIS, INC. PROPRIETARY INFORMATION
AND INVENTIONS AGREEMENT

1. The following is a complete list of all inventions or improvements relevant to the subject matter of my employment by Synopsis, Inc. (the "Company") that have been made or conceived or first reduced to practice by me alone or jointly with others prior to my employment by the Company that I desire to remove from the operation of the Company's Proprietary Information and Inventions Agreement.

_____ No inventions or improvements

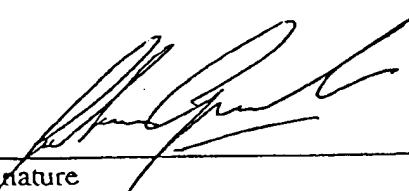
_____ See below

X Additional sheets attached

2. I propose to bring to my employment the following materials and documents of a former employer:

X No materials or documents

_____ See below



Signature

Lukas van Glnneken,

Print or Type Name

ATTORNEYS AND
CONSULTANTS ONLY

SY000006

EXHIBIT B
TO
SYNOPSYS, INC. PROPRIETARY INFORMATION
AND INVENTIONS AGREEMENT

Section 2870. Application of provision providing that employee shall assign or offer to assign rights in inventions to employer.

(a) Any provision in an employment agreement which provides that an employee shall assign, or offer to assign, any of his or her rights in an invention to her or her employer shall not apply to an invention that the employee entirely on his or her own time without using the employer's equipment, supplies, facilities or trade secret information except for those inventions that either:

(1) Relate at the time of conception or reduction to practice of the invention to the employer's business, or actual or demonstrably anticipated research or development by the employer;

(2) result from any work performed by the employee for the employer.

(b) To the extent a provision in an employment agreement purports to require an employee to assign an invention otherwise excluded from being required to be assigned under subdivision (a), the provision is against the public policy of this state and is unenforceable.

ATTORNEYS AND
CONSULTANTS ONLY

SY000007

The following is a complete list of all inventions or improvements relevant to the subject matter of my employment by Synopsys, Inc. (the Company) that have been made or conceived or first reduced to practice by me alone or jointly with others prior to my employment by the Company that I desire to remove from the operation of the Company's Proprietary Information and Inventions Agreement.

[0] "Efficient orthonormality testing for synthesis with pass transistor selectors" by M. R. C. M. Berkelaar and -, accepted at the International Workshop on Logic Synthesis, June 1995.

[1] "Timing Verification and Optimization for the PowerPC Processor Family", by R.E. Mains, T. A. Mosher, - and R.F. Damiano, in: Proc. Int. Conf. on Computer Design, pp.390-393, Boston, Oct. 10-12, 1994.

[2] "In the driver's seat of BooleDozer" by D. Brand and R.F. Damiano, -, A. D. Drumm, in: Proc. Int. Conf. on Computer Design, pp. 518-521, Boston, Oct. 10-12, 1994.

[3] "Grammar-based optimization of synthesis scenarios" by A. Kuehlmann and -, in: Proc. Int. Conf. on Computer Design, pp. 20-25 Boston, Oct. 10-12, 1994.

[4] "Tuning of logic synthesis scenarios" by - and A. Kuehlmann, Workshop notes of the Int. Workshop on logic synthesis, paper P7c, Tahoe City, May 23-26, 1993.

[5] "Fanin ordering in multi-slot timing" by -, Proc. Int. Conf. on Computer Design, pp. 44-47, Cambridge, Oct. 11-14, 1992.

[6] "The complexity of adaptive annealing" by R. H. J. M. Otten and -, Proc. Int. Conf. on Computer Design, pp. 404-407, Cambridge, Sept. 17-19, 1990.

[7] "Buffer placement in distributed RC-tree networks for minimal Elmore delay" by -, Proc. Int. Symp. on Circuits and Systems, pp. 865-868, New Orleans, May 2-5, 1990.

[8] "Optimal slicing of plane point placements" by - and R. H. J. M. Otten, Proc. European Design Automation Conf. pp. 322-236, Glasgow, March 12-15, 1990.

[9] The annealing algorithm by R. H. J. M. Otten and -, ISBN 07923-9022-9, Boston:Kluwer, 1989.

[10] The predictor-adaptor paradigm - automation of custom layout by flexible design by -, Ph.D. thesis, ISBN 90-9002703-3, Eindhoven, 1989.

[11] "Doubly folded transistor matrix layout" by - and J. T. J. van Eijndhoven, A. H. C. M. Brouwers, Digest Int. Conf. on Computer Aided Design, Santa Clara, Nov. 7-10, 1988.

[12] "Stop criteria in simulated annealing" R. H. J. M. Otten and -, Proc.

Int. Conf. on Computer Design, pp. 549-552, Port Chester, Oct. 3-5, 1988.

[13] "An inner loop criterion for simulated annealing" by - and R.H.J.M. Otten, Physics letters A, 130:429-435, 1988.

[14] "Soft Macro Cell generation by two dimensional folding" by - and J. T. J. van Eijndhoven, P. R. M. van Teeffelen, T. J. Deckers, Proc. Int. Symp. on Circuits and Systems, pp. 727-730, Espoo, June 1988.

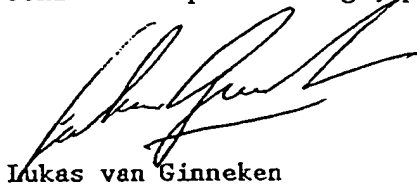
[15] "Gridless routing of general floor plans" by - and J. A. G. Jess, Digest Int. Conf. on Computer Aided Design, pp. 30-33, Santa Clara Nov. 9-12, 1987.

[16] "Wire planning for stackable designs", by R. K. Brayton, C. L. Chen, J. A. G. Jess, R. H. J. M. Otten and -, Proc. Int. Symp. on VLSI technology, pp. 269-273, Taipeh, May 13-15, 1987.

[17] "Global wiring for custom layout design" by - and R. H. J. M. Otten, Proc. Int. Symp. on Circuits and Systems. pp. 207-208, Kyoto, June 5-7, 1985.

[18] "Floor plan design using simulated annealing" by R. H. J. M. Otten and -, Digest Int. Conf. on Computer Aided Design, pp. 96-98, Santa Clara, Nov. 1984.

[19] "Stepwise layout refinement" by - and R. H. J. M. Otten, Proc. Int. Conf. on Computer Design, pp. 30-36, Port Chester, Oct. 8-11, 1984.



Lukas van Ginneken

ATTORNEYS AND
CONSULTANTS ONLY

SY000009

DECLARATION OF ROBERT DAMIANO

I, Robert Damiano, declare as follows:

1. The following statements are based on my personal knowledge. If called upon to testify, I could and would competently testify as to the matters set forth herein.
2. I am an employee of Synopsys, Inc. My present position is Vice-President of the Advanced Technology Group. In September 1996, my position at Synopsys was Director in the Advanced Technology Group. Lukas van Ginneken, who was also employed at Synopsys during that time, worked on one of my projects.
3. On or about September 9, 1996, I received an email from Lukas van Ginneken that included a draft patent application. A true and correct copy of such email is attached hereto as Exhibit A.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of Declarant:

Robert Damiano

Declarant's Signature:

Robert Damiano

Date:

28 January 2005

Exhibit A

[Document Follows]

TO: robertd@synopsys.com
From: Lukas van Ginneken <lukas@synopsys.com>
Subject: patent
Date: 1996-09-09 22:07:54 GMT

SYNOPSYS CONFIDENTIAL

APPLICATION FOR UNITED STATES PATENT

in the name of

LUKAS PAUL PIETER PEPIJN VAN GINNEKEN

of

SYNOPSYS, INC.

for

METHOD FOR ACHIEVING TIMING CLOSURE OF DIGITAL NETWORKS

AND

METHOD FOR AREA OPTIMIZATION OF DIGITAL NETWORKS UNDER TIMING CLOSURE

class 364/489

BACKGROUND OF THE INVENTION

This application relates to a method for achieving timing closure of digital networks consisting of structuring and mapping and a method for area optimization of digital networks using placement and sizing, while maintaining timing closure.

(Prior Art)

Figure 1 shows the conventional approach to digital network synthesis.

Digital network synthesis is a process in which computer programs optimizes digital networks. At the beginning of the synthesis process, a human designer 110 specifies a design 111 at a high level of abstraction using a high level design specification language, such as Verilog or VHDL.

In step 101 the abstract network specification of the design is transformed into an unmapped digital network representation in memory by a computer program, such as HDL Compiler, available from Synopsys Inc. of Mountain View, California.

In step 102 Logic synthesis algorithms optimize the network by changing the structure of the network without changing the function of the network.

In step 103, logic network synthesis algorithms map the abstract network representation to cells from the library.

Step 104 optimizes the size of the cells. The size of the cell, together with the load of the cell determines the delay of the cell. The sizing algorithm adjusts the sizes of the cells. Changing the sizes of the cells affects delay and area, and the sizing algorithm manipulates the cell sizes so as to minimize delay and area.

Steps 102, 103 and 104 are performed by a computer program, such as "Design Compiler" TM available from Synopsys Inc., of Mountain View, California.

Step 105 determines the placement of the cells on the chip. Placement algorithms attempt to keep the length of the nets short, as longer nets need more area on the chip and the increased net load of longer nets will make the network slower. The network remains unchanged during the placement.

Finally step 106 determines the exact routing of the nets on the chip. Steps 105 and 106 are done by a computer program, such as "Cell Ensemble" TM available from Cadence Inc. of San Jose, California.

(Problems with prior art)

The major problem with the conventional approach is that the net length and hence the cell delay is not known until after placement. Before placement, net length must be estimated. This is usually done with an estimation function or table which gives the load of a net based on its fanout. Experience has shown that it is very difficult to estimate the length of the nets accurately. Essentially net length behaves as a random variable.

The result is unpleasant surprises after placement step 105. Some nets turn out to be longer than expected, and because of the longer delays the timing constraints are not met. Timing closure is not certain until after step 105.

If timing closure is not achieved the options the designer has are expensive and unreliable. He may choose to fix the design manually, which is difficult and time consuming, because the automatically optimized network is hard to understand. He may choose to change his HDL specification and repeat the synthesis process. Again timing closure will not be certain until after placement, which means that the entire process needs to be traversed before the designer knows if his HDL changes were successful.

A common method of dealing with inaccurate net load estimates is to use net load estimates which are considerably larger than accurate estimates. This causes the sizes of the cells to be considerably larger than necessary but reduces the probability of not meeting the timing constraints after placement. Clearly using cells with sizes which are larger than necessary is wasteful in both silicon area and power consumption. The chips thus synthesized will be larger, cost more to produce and use more electrical power than necessary.

A second problem with the conventional approach is that the effect

of synthesis decisions is hard to calculate. Performing timing analysis during optimization is very time consuming, and accounts for most of the run time of conventional synthesis systems.

In step 103 it is difficult to take decisions based on delay without knowing load and size of the cells as well.

In step 104, changing the size of a cell affects the loads of the fanin cells, and thus the delay of the fanin cells. In more complex delay models, which take into account the transition time of the signals, also the delay of the fanout cells is affected. Usually the size parameter cannot have any arbitrary value: Because the library of cells has been designed before the network synthesis started, only a few sizes are available. 3 or 4 sizes per cell is common. This makes it harder to find a good solution.

In step 105 the placement program will modify the net lengths. Depending on which location was chosen for each cell, the length of each net can be different. As the length differs, the capacitive load of the net changes, and as a result, the delay of the cell driving the net changes. Therefore the delays which were carefully optimized during the logic synthesis, are very different after placement, and the optimization of the network is not very good.

Much of the progress in the state of the art can be characterized as increased integration. This is represented in figure 1 as various feedback paths, which repeat and alternate steps. The general direction has been towards programs which do structuring, mapping, sizing and placement simultaneously. It has lead to increasingly complex software systems which are slow and difficult to design and maintain.

Iterating between placement and sizing has been especially hard to execute because placement programs are not sold by the same design automation software vendors as logic synthesis programs. Also they are not run by the same users; the logic synthesis program is often run by the designer, who also wrote the HDL specification. The placement program is often run by the silicon chip manufacturer, after the design is considered complete.

SUMMARY OF THE INVENTION

(objectives)

It is an object of the present invention to achieve timing closure as quickly as possible in the synthesis process. This will give the human designer early warning if the design is infeasible, because it is over constrained.

The invention achieves this object by

- speeding up the algorithms in the synthesis process
- performing the feasibility check early in the synthesis process, that is, before placement, instead of after placement.
- maintaining feasibility throughout the remainder of the synthesis process, so that it can be guaranteed to succeed and can be executed automatically.

The present invention overcomes the problems of the conventional approach by not choosing a size for a cell at all. Rather than choosing a default size, as conventional methods do, we choose a delay and let the size implicitly be whatever it needs to be to meet that delay.

In the conventional method of optimization, the structure, mapping, size and placement are chosen to optimize delay and area. In our formulation of the problem, we choose the structure, mapping, delay and placement, to optimize size and area. In our formulation, size only affects the area, so area only remains as an optimization goal.

The present invention speeds up the programs by simplifying or eliminating timing analysis. While optimizing network delay in order to achieve timing closure, the delays are constant, which will speed up delay calculation. Also, since changes to the network do not change the delay of the cells, the amount of recalculation is drastically reduced. While optimizing area after timing closure has been achieved, timing analysis is not needed, as the delay of the cells does not change. The one step where timing analysis using complex delay models is necessary is in the stretching step. Here too, library design rules are taken into account.

The present invention maintains timing closure after it has been achieved by adjusting the size of the cell during or after placement. The adjustments compensate for the fact that the placement algorithm can assign different net lengths to different nets and that these lengths are difficult to predict before placement.

(language of the main claims)

In accordance with the purpose of this invention, as embodied and broadly described herein, the invention is a method for the structuring and mapping of an unmapped digital network comprising the following steps:

- a) structuring of the digital network, using network slack as an optimization goal, where network slack is calculated assuming that the delay of the cells of the network is constant with respect to load.
- b) mapping of the cells of the network, using network slack as an optimization goal, where network slack is calculated assuming that the delay of the cells of the network is constant with respect to load.
- c) estimation of the area of the network based on net load

In further accordance with the purpose of this invention, as embodied and broadly described herein, the invention is a method for the placement and sizing of cells of a mapped digital network, the method comprising the steps of:

- d) Choosing a target delay for each cell.
- e) Computing the network slack using the target delays.
- f) Placement of the cells of the network.
- g) Sizing of the cells of the network such that the network meets the network slack as computed by step b).

In further accordance with the purpose of this invention, as embodied and broadly described herein, the invention is a method for the placement of the cells of a digital network, the method comprising the steps, of:

- f1) The calculation of net weights that reflect the change of network area due to sizing as a function of net length.
- f2) Placement of the cells of the network where the weighted network net length is used as a placement objective, the weighted network net length being the sum of the weighted net lengths of all nets, each net length being

multiplied by a weight,

Objects and advantages of the invention will be set forth in part in the description which follows and in part will be obvious from the description or may be learned by practice of the invention. The objects and advantages of the invention will be realized and attained by means of the elements and combinations particularly pointed out in the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute part of this specification, illustrate several embodiments of the invention and, together with the description, serve to explain the principles of the invention.

Fig. 1 is a flow chart showing the flow of the conventional method.

Fig. 2 is a flow chart showing the flow according to the present invention.

Fig. 3 is a block diagram of a computer.

Fig. 4 is a schematic diagram of a digital network.

Fig. 5 is a schematic diagram of the electronic and the physical implementations of a cell.

Fig. 6 is a timing diagram illustrating the concept of slack and other timing concepts.

Fig. 7 is a graph showing the relationship between the delay of a cell, the size of a cell and the load of a cell.

Fig. 10 is a

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Reference will now be made in detail to the preferred embodiments of the invention, examples of which are illustrated in the accompanying drawings. Wherever possible, the same reference numbers will be used throughout the drawings to refer to the same or like parts.

(what insight led to this invention?)

(Software patent application)

The preferred embodiment of the present invention is as one or more computer programs. The digital network, its elements and attributes, exist during the process of the method only as data structures in the memory of the computer. Methods in prior art are known to persons of ordinary skill in the art to convert the design data in memory eventually to an actual physical implementation of the network.

(Computer system)

Figure 1 is a block diagram of a computer system 100 in accordance with

the present invention. Computer system 100 includes a central processing unit 101, bus 102, memory 103, input device 104 and output device 105. It will be understood by a person of ordinary skill in the art that computer system 100 can also include numerous elements not shown in the figure for the sake of clarity, such as disk drives, tape drives, mice, printers, network connections, additional CPUs, etc. Memory 103 contains a program 107, which embodies the invention, and a data structure representation of the network 106.

(Network terminology)

Figure 2 is a schematic diagram of a digital network. Digital *network* 200 is composed of a plurality of *cells* 205, 206, 207, 208, 209, connected by a plurality of *nets*. Each cell (e.g. 208) has one or more inputs 212, 213, and a single output 214. Each net is connected to one output and one or more inputs. The cells can be combinational *gates* 207, 208, 209, whose function is represented as an expression in the Boolean algebra, using AND, OR and NOT operators, or the cells can be registers 205, 206. All feed back loops 210, 211 in the network contain at least one register. Cells whose outputs are connected to the inputs of a cell are called the fanin of the latter cell. Cells whose inputs are connected to the output of a cell are called the fanout of the latter cell.

(Computation)

The digital network performs a logic *function* by processing digital binary input data in a number of cycles. The input data is presented to the network on its *primary inputs* 201, 202, and the result of the computation of the network function is presented at the *primary outputs* 203, 204, of the network. The computation of the function takes one or more cycles. During each cycle the gate functions are calculated. The results are stored in the registers for use in the next cycle. The data flow is from inputs to outputs and will be assumed to be pictured as going from left to right throughout this text.

(Mapping)

Each cell can be *mapped* to a *book* in which case a electronic realization in terms of transistors has been chosen, and physical attributes such as delay are known. If a cell is not mapped to a book it is *unmapped*, in which case no electronic realization has been selected, and the function of the cell is only known in abstract terms, such as Boolean algebraic expressions. A *library* of books is designed in advance, before synthesis starts. These books are of generic types, and can be used to build arbitrary designs.

(Size)

The cell has a delay, an area and its input pins have an input pin capacitance. The *size* of the cell is a multiplier which is applied to equally to all transistor channel widths in the electronic circuit of the cell. Thus size of a cell is a scale factor which is used to scale its output load driving capability (see below), its area, and its input pin loads.

(Timing constraints)

Each primary input or primary output has an associated delay,

called the *input delay*, respectively *output delay*, which represents delays external to the network. The network needs a certain amount of time to perform one cycle, called the *cycle delay*. Together the cycle delay, the input delays and the output delays form the timing constraints of the network. Meeting the timing constraints is called *timing closure* and it is a major objective of the synthesis process.

(Timing)

The delay of a path is measured as the sum of the gate delays over said path from begin point to end point. The cycle delay is the maximum of all path delays. Primary outputs and register inputs are timing path end points. Primary inputs and register outputs are timing path begin points.

(Slack)

The *arrival time* of the data at a gate is computed by taking the maximum arrival time of its fanin cells each increased by the delay from the input pin to the output pin. The arrival times can be computed by traversing the network from left to right, that is, from timing begin points to timing end points. Similarly the *required time* of the data at a gate can be computed by traversing the network from end points to begin points. The required time is the minimum required time of its fanout cells, each reduced by the delay from the input to the output pin of that fanout cell.

The difference between the required time and the arrival time is the *slack*. If the arrival time is smaller than the required time, the timing constraints are met, and the slack is positive. If the arrival time is larger than the required time, the timing constraints are not met, and the slack is negative. The arrival time and required time may be different depending on whether the data is zero (0) or one (1). There also may be multiple arrival times and multiple required times to model a variety of timing constraints. All slacks can be summarized as a single worst slack number, called the *network slack*. Timing closure is achieved if the network slack is non-negative.

(Delay model)

The delay D of a gate depends on many factors, among them its function, its size S and the capacitive load C of the gate. The delay may be different for different inputs of the gate and it may be different for the falling and the rising transition. It is important to note that the dependency on size and load can be captured as the dependency on a single parameter C/S , and the delay D is non-negative and monotonically increasing with C/S .

$$\begin{aligned} C &> 0 \\ S &> 0 \\ D &= f(C/S) \\ f(C/S) &> 0 \\ f'(C/S) &> 0 \end{aligned}$$

Figure 4 illustrates the relationship between the three variables, size S , delay D and capacitive load C . Each of the three planes shows the relationship between two variables, while the third variable is constant, (not necessarily zero).

(Detailed description)

The preferred embodiment of the invention is a software program that can be stored in the memory of a computer, and can be executed by the central processing unit of the computer so that the computer performs the method described herein.

The software program consists of many parts or subprograms which together perform the method described in this invention. The essence of the invention is that logic synthesis is done in a size independent way, and that sizes are determined after placement, and that we guarantee that that the delay numbers before placement can be met by sizing after placement.

The preferred embodiment consists of three parts: (See fig...) to wit

- a) The logic synthesis program
- b) The placement program
- c) The sizing program

- analyze the library
- read logic
- library independent optimization.

The first step is to analyze the library that will be used for logic synthesis. The library contains the cells that will be used to implement the logic function. Contrary to the standard method of performing logic synthesis, we will assume that each cell can be sized by a continuous, positive real variable S , which increases both the load driving capability of the cell and the area linearly. In other words, the area of a cell is $S \cdot A$ and the delay of a cell is $D = f(C/S)$.

The library analysis will determine a good value for C/S for each cell in the library. Using this value, it determines a constant delay for each gate.

(choice of C/S)

Since the library analysis is not dependent on the actual network being synthesized, library analysis can be performed before beginning the synthesis process. We will now continue to describe the actual automatic synthesis process, beginning at the with reading the design. The design is expressed in a high level design specification language, for example VHDL or Verilog, and is syntactically parsed and transformed into a logic network representation by a computer program, such as HDL Compiler, available from Synopsys Inc. of Mountain View, California.

(library independent optimization)

Initially the network is library independent and library independent optimizations are performed. Mostly these optimizations change the structure of the network, and the Boolean functions of the cells, without changing the overall function of the network. The types of optimizations that should be performed are behavioral optimizations, such as resource sharing, sequential optimizations such as retiming, algebraic optimizations such as kernel extraction and Boolean optimizations such as redundancy removal. There is a large amount of literature on how each of these classes of optimizations can be performed.

(mapping for delay)

Following the library independent optimisations the network is mapped to a library of cells. This means that the logic functions of the cells are implemented with actual cells from the library. During this process the A large body of literature exists already on the subject of mapping digital networks. The preferred embodiment would use a previously published algorithm such as

(post mapping optimizations)

Due to restrictions in run time, it is impractical to explore the entire design space during the mapping algorithm. Necessarily, the mapping algorithm has to ignore many possible solutions because either they are unlikely candidates or they are very similar (but not identical) to other, considered solutions. In addition, in the constant delay approach, it is easy to evaluate the impact on timing of synthesis decisions, but it is much harder to evaluate the impact on the total network area. Therefore the mapping algorithm necessarily cannot accurately optimize area.

(pin swapping)

An example of candidate mappings which are not explored during mapping because they are too similar to other mappings follows here: Often gates have several pins which are functionally interchangeable. For instance for a 4 input NAND gate, there are $4! = 24$ possible permutations for the 4 input pins. Usually these pins are not equally fast, because of an inherent asymmetry of the electronic circuit. Because the differences are small, it is not worthwhile to consider all of these different of different permutations during mapping. It is more efficient to pick one arbitrarily and to select the best permutation of the inputs after mapping.

(structuring - boundary move)

Using constant delay it is considerably easier to predict the effect of a change to the network than with the conventional delay models. This can be used to do timing optimization by means of restructuring after technology mapping has been done. For example, we can use the "boundary move" transformation to reduce the number of levels in the logic in the mapped network. The boundary move transform, illustrated in fig x, reduces the number of levels by bringing connection x forward. To make the change legal it is necessary that gates x, y and z are fanout free. If not, they must be made fanout free by making a copy. In the conventional approach to logic synthesis, copying logic will increase the load on gates x, x, x and therefore increase the delay. To predict if the transformation will improve delay, or hurt delay, it was necessary to run a complete static timing analysis with accurate delay models. If the change actually worsened the delay, then the change would be undone.

In the constant delay model approach, the effect of this change can be easily predicted. Note that changes in loads do not affect delay. The only change that affects delay, is the change of the fanin of gate xxx. The delay can easily be predicted by simple addition of gate delays.

(area estimation)

To perform area optimization it is necessary to calculate the sizes of the cells. The sizes can be calculated in a straightforward manner from the loads. The loads are calculated by adding the net load and the pin load. The net load consists of the load of the net, which can be estimated using a conventional net load model, plus any other fixed load, such as the load of a primary output. The pin load is not fixed, that is, the load of an input pin depends on the size of the cell. This creates a dependency: To calculate the load of a cell, we need to calculate the size of its fanout cells. Therefore the algorithm starts calculating as far downstream as possible, and traverse the network in a direction opposite to the flow of data. In a combinational network this can be achieved by starting at the primary outputs and traversing the network in a leveled order towards the primary inputs. In a sequential network there may be one or more loops, resulting a cyclic dependency: there is no rightmost cell. In this case the computation can start anywhere in the cycle, and repeats the cycle several times, until the capacitances converge and the error is sufficiently small.

It is possible that this iteration will not converge and that the capacitance will increase in every iteration, by progressively larger amounts. This situation is detected by requiring the increment to be smaller than a preset maximum after a fixed number of iterations. The iteration does not converge if the network is an infeasible solution: The current network cannot be expected to work at this speed because its gain is too small. Changes need to be made to the network to increase the gain, which will usually mean increasing the delay of the network as well.

After the loads have been calculated the size can be calculated by dividing the actual load by the predetermined typical load. The input capacitance can be calculated by multiplying the unit gate input capacitance by the size. The ratio of these numbers is the size of the gate. The size is a scale factor, which can be applied to the area of the gate, to give the area of the sized gate. The area of the network can be estimated as the sum total of the areas of the sized gates, plus the net area as estimated from the total length of all nets.

(net weights)

Various algorithms, such as buffering and placement optimize the network by manipulating the loads in the network. Placement optimizes the net length, which directly related to the net load, and buffering reduces the load on a cell by adding extra delay. These algorithms can benefit from a more efficient calculation of the effect that changing the load of a cell has on sizing. We can do this by calculating single parameter per net, called the net weight, which represents the sensitivity of the total area of the network with respect to the load on that net.

This net weight can be calculated in a manner that is very similar to the calculation of the pin loads during the area calculation above. Starting at the primary inputs, the net weight of the first (left-most) gate is equal to its area per unit load. The net weights of the other cells can now be calculated with a recurrence relation traversing the network from left to right.....

to the calculation of the loads. Starting at the primary inputs the network to 0. Set the net load on the net in question to 1. Perform the iteration and calculate the area as described in the previous section.

Since all calculations are linear, the effects simultaneous changes in loads of several nets can be superimposed, that is, added together.

(buffering)

The next step in the synthesis process is the buffering step. The buffering algorithm adds buffers to the network guided by the timing analysis and the area analysis. In the constant delay model, the buffers have a fixed delay, and thus the impact of inserting a buffer is easily determined by subtracting the delay of the buffer from the slack. Thus the effect of adding a buffer on delay is always negative: a buffer always adds delay, never reduces delay. The main effect of adding a buffer is to save area because the source gate can be smaller because the it's load is smaller. The effect on area and gain can be determined by area analysis. Net weights.....

The buffering algorithm works as follows: First it finds locations in the network where a buffer can be added without increasing the network delay. This is done by subtracting the delay of the buffer from the local slack, to give the predicted slack after buffer insertion. If the predicted slack is larger than the network slack, then a buffer can be inserted without increasing the network delay. Next we have to calculate the reduction in load of this net, and check that area that is added by adding the buffer does not exceed the area saved by sizing down the source gate. The area added by inserting the buffer is simply the area of the buffer times its size, where the size is determined by the load on the buffer divided by the typical load of the buffer. The area saved by inserting the buffer can be calculated by first calculating the change in load due to the insertion of the buffer: some sinks are removed, the input load of the buffer is added, and the net load estimate may change as a result of the number of fanouts of the net changing. Using the net weight, we can estimate the impact on the network area. If the impact is positive (reduced area) the buffer is inserted. After the buffer is inserted, the capacitances need to be updated in the fanin cone of the buffer, while the net weights need to be updated in the fanout cone of the buffer.

(stretching)

The next step is the process of "Stretching" and "Compressing" the delays of the individual gates to meet the timing constraints. Gates which are on long paths which do not meet the delay constraint are "compressed" until the path does meet the timing constraint. Gates on the short paths which easily meet the timing constraints are "stretched". Gates with stretched delays require less area for the same load. In this step the delay of the gates is traded against the gain of the gates. When the delay of a gate decreases so does the gain of the gate. It is important that there is enough gain in the network.

Note that

this can be done entirely independent of the sizes of the gates. Sizes of gates are not determined until much later in the process.

The stretching algorithm has two phases. In the first phase it will compress the delays of cells on long paths to meet timing constraints. In the second phase it will stretch the delays of the

gates on short paths to save area. For the purpose of stretching and compressing registers can usually be considered to be part of a path which they originate, but not of a path that they terminate. The stretching algorithm considers the cells on a path by path basis, processing the path with the smallest slack first. The delay of each cell on the path is adjusted by an amount which is equal to the slack divided by the number of stages on the path. After a cell has been adjusted, it becomes "locked" and its delay cannot be changed by the stretching algorithm. Stages which are locked are not counted when calculating the adjustments. For the stretching phase, the algorithm continues as above. In this phase the delays of the cells are increased, not decreased. The path that we work on is not the worst path, but it is the worst path with a slack greater than 0. (All other paths now have a slack of 0).

(Rule based stretching)

(incremental rule based)

(placement)

In our process a conventional placement method is augmented to optimize the area of the placed network. All placement methods known work by gradual refinement of the placement. Periodically, during the placement process, we recalculate the estimated net lengths, using the most recent, accurate placement information. From the net lengths, it calculates the sizes of the cells in the network. (See area estimation). The updated sizes can then be used for further placement and for more accurate net length calculations.

Placement primarily manipulates the lengths of the nets. Using the net weights, the area of the network can efficiently be estimated.

(final or discrete sizing)

Other embodiments will be apparent to those skilled in the art from consideration of the specification and practice of the invention disclosed herein. It is intended that the specification and examples be considered as exemplary only, with a true scope of the invention being indicated by the following claims.

WHAT IS CLAIMED IS:

(CONSTANT DELAY SYNTHESIS)

1. A method for the structuring and mapping of an unmapped digital network comprising the following steps:

- a) structuring of the digital network, using network slack as an optimization goal, where network slack is calculated assuming that the delay of the cells of the network is constant with respect to load.
- b) mapping of the cells of the network, using network slack as an optimization goal, where network slack is calculated assuming that the delay of the cells of the network is constant with respect to load.
- c) estimation of the area of the network based on net load

(typical load)

3. The method of claim 1, the delay of a book is used for the delay of each cell, the delay of the book being chosen by choosing a load size ratio C/S for each book, which is independent of the network.

4. The method of claim 3, where the delay of a book is also determined by the choice of the input transition time, which is independent of the network.

(continuous buffering assumption)

5. The method of claim 3,

where a parameter C/S for each book

is chosen to have the largest

possible value such that a long chain of cells of identical books each cell in the chain having identical value of parameter C/S, said chain cannot have simultaneously improved delay and improved gain by adding a buffer at some point to the same chain, even when the parameter C/S is chosen optimally after adding the buffer.

(buffering)

6. The method of claim 1, with the additional step of buffer insertion before step c), the buffers being inserted on paths with positive slack, as determined by subtracting the delay of the buffer from the slack of the path.

(more buffering)

7. The method of claim 6, where the buffer is inserted if area is saved.

8. The method of claim 7, where the area savings are estimated using net weights which reflect the change of network area due to sizing as a function of net length.

9. The method of claim 8, where the calculation of the net weights

is performed by starting at the primary inputs and traversing the network in the direction of the data flow, while calculating the net weight of a cell by summing over the fanin of the cell the product of the net weight of the fanin cell divided by the gain plus the cell's area/load sensitivity.

as $N_i = \sum_j W_j / g_{ij}$

(stretching)

10. The method of claim 1, where step a) is comprising of:

a1) choosing a delay for each book

a2) the delay of the cell assuming the delay of its book

a2) adjusting the delay of each cell based on the slack

11. The method of claim 10, where in step a2) the delay of each cell is adjusted equally among the stages which have the same slack.

12. The method of claim 10, where in step a2) the delay is adjusted on each path, such that the slack of each path becomes 0.

(globally optimal mapping)

13. The method of claim 1, where in step b) the mapping is performed in two steps:

b1) a traversal of the network from

primary inputs and registers from left to right, while choosing at each cell the fastest matching books from all available matching books, using the constant delays of the books and the fastest arrival times of the fanins

of the matching book.

b2) a traversal of the network from right to left, while choosing at each cell the fastest matching book from the candidates selected during the previous traversal.

(Area estimation)

14. The method of claim 1, where in step c) consists of the following steps:

- c1) estimation of the net length based on the number of fanout cells.
- c2) estimation of the capacitive load of the cells using the net length
- c3) calculation of the sizes from the capacitive load.
- c4) calculation of the network area by summation of the product of the book area times the cell size.

(sizing algorithm)

15. The method of claim 14, where step c2) is performed by starting at the primary outputs and traversing the network in the direction opposite to the data flow, while calculating the load of a cell, by summing over the fanout of the cell, the product of the load of the fanout cell divided by the gain plus the net load of the cell.

(iterate till convergence)

16. The method of claim 15, where the network has loops, the traversal is started at primary outputs and arbitrary cells in the loop, and the calculation is repeated until convergence.

17. The method of claim 16, where the traversal is started at primary outputs and registers.

(Area optimization)

18. The method of claim 1, where steps a) and b) also use network area as an optimization goal, in addition to network delay, the network area being estimated as in step c).

(retiming)

19. The method of claim 1, where the structuring step is preceded by a retiming step, where registers are moved in the network, while preserving the function of the network, and where change to the network do not affect the delay of the individual cells.

(I could produce a bunch more of these sort of claims)

(TIMING CLOSURE)

20. A method for the placement and sizing of cells of a mapped digital network, the method comprising the steps of:

- d) Choosing a target delay for each cell.
- e) Computing the network slack using the target delays.
- f) Placement of the cells of the network.
- g) Sizing of the cells of the network such that the network meets the network slack as computed by step b).

(sizing for better placement)

21. The method of claim 20, where step f) is performed in gradual steps, each step being followed by a sizing step g)

(partitioning)

22. The method of claim 21, where step f) is performed by repeated partitioning steps, partitioning the cells in the network into two or more groups, each group being assigned to an subdivision of the plane, alternating the partitioning steps with sizing steps, essentially similar to step g)

(iterative improvement)

23. The method of claim 21, where step f) is performed by choosing an arbitrary initial location in the two dimensional plane for each cell, the placement being optimized by repeatedly changing the location of one or two cells at a time, while performing a sizing step, essentially similar to step g) after each location change.

(sizing)

24. The method of claim 5, where step g) is consists of the following steps:
g1) calculation of the net length based on the available placement information
g2) calculation of the capacitive load of the cells using the net length
g3) calculation of the sizes from the capacitive load.

(sizing algorithm)

25. The method of claim 24, where step g2) is performed by starting at the primary outputs and traversing the network in the direction opposite to the data flow, while calculating the load of a cell, by summing over the fanout of the cell, the product of the load of the fanout cell divided by the gain plus the net load of the cell.

(iterate till convergence)

26. The method of claim 10, where the network has loops, the traversal is started at primary outputs and arbitrary cells in the loop, and the calculation is repeated until convergence.

27. The method of claim 26, where the traversal is started at primary outputs and registers.

(cell generation)

28. The method of claim 20, where following step g) the layout of the cells is generated automatically to yield the exact transistor sizes calculated by step g)

(discrete sizing)

29. The method of claim 24, where step g3) consists of selecting the most suitable size from a limited set of available sizes.

(Weighted Placement)

30. A method for the placement of the cells of a digital network, the method comprising the steps, of:

f1) The calculation of net weights that reflect the change of network area due to sizing

as a function of net length.

f2) Placement of the cells of the network

where the weighted network net length is used as

a placement objective, the weighted network net length being the sum

of the weighted net lengths of all nets, each net length being multiplied by a weight,

(Calculation of net weights)

31. The method of claim 30, where step f1)

is performed by starting at the primary inputs and traversing the network in the direction of the data flow, while calculating the net

weight of a cell by summing over the fanin of the cell the product of the net weight of the fanin cell divided by the gain plus the cells area/load sensitivity.

as $W_i = \sum_j W_j / g_{ij}$

(iterate till convergence)

32. The method of claim 31, where the network has loops, the traversal is started at primary inputs and arbitrary cells in the loop, and the calculation is repeated until convergence.

33. The method of claim 32, where the traversal is started at primary inputs and registers.

(Weighted Placement for power)

34. A method for the placement of the cells of a digital network, the method comprising the steps, of:

f1) The calculation of net weights that reflect the change of network power due to sizing

as a function of net length.

f2) Placement of the cells of the network

where the weighted network net length is used as a placement objective, the weighted network net length being the sum of the weighted net lengths of all nets, each net length being multiplied by a weight,

(Calculation of net weights)

35. The method of claim 30, where step f1)

is performed by starting at the primary inputs and traversing the network in the direction of the data flow, while calculating the net weight of a cell by summing over the fanin of the cell the product of the net weight of the fanin cell divided by the gain plus the cells power/load sensitivity.

as $W_i = \sum_j W_j / g_{ij}$

(iterate till convergence)

36. The method of claim 31, where the network has loops, the traversal is started at primary inputs and arbitrary cells in the loop, and the calculation is repeated until convergence.

37. The method of claim 32, where the traversal is started at primary inputs and registers.

38. The method of claim 35, where the power/load sensitivity of a cell is calculated as the product of the switching frequency at that cell times the square of the voltage times the capacitance, where capacitance is calculated as the sum of the net load plus the internal capacitance of the cell, scaled with its size.

ABSTRACT

A method for the design of digital networks consisting of a plurality of cells. The invention uses constant delays during logic synthesis and sizes the cells after placement so as to meet the cycle delay predicted before synthesis. The method chooses a constant delay before logic synthesis and guarantees that it can maintain this delay after placement by means of sizing. Thus it overcomes the unpredictable effects of placement on the cycle delay. The invention also describes methods for choosing the sizes of the gates and a method for inserting buffers.

Synopsis Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 2</u></p> <p>The major problem with the <i>conventional approach</i> is that the net length and hence the cell delay is not known <i>until after placement</i>.</p>	<p><u>'446 PATENT AT 1:46-47</u></p> <p>Thus, under the <i>conventional</i> design approach, timing closure is not certain <i>until after placement</i>.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 2-3</u></p> <p>Before placement, <i>net length</i> must be <i>estimated</i>. This is usually done with <i>an estimation function or table which gives the load of a net based on its fanout</i>. Experience has shown that it is very difficult to <i>estimate</i> the length of the nets <i>accurately</i>.</p>	<p><u>'446 PATENT AT 1:37-40</u></p> <p>While <i>net lengths</i> have been <i>estimated</i> prior to placement by use of <i>an estimation function or table which gives the load value of a net based on</i> the number of <i>fanout</i> gates, this <i>estimation</i> function is usually <i>inaccurate</i>.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 3</u></p> <p>The result is unpleasant surprises <i>after placement</i> step 105. <i>Some nets turn out to be longer than expected</i>, and because of the <i>longer delays</i>, the <i>timing constraints</i> are not met. <i>Timing closure is not certain until after</i> step 105.</p>	<p><u>'446 PATENT AT 1:41:46</u></p> <p>This difficulty in accurately predicting net lengths leads to unpredictable delay effects <i>after cell placement</i> occurs. For example, <i>some nets turn out to be longer in length than expected</i>. These longer nets cause <i>longer delays</i> which prevent satisfaction of <i>timing constraints</i> in the digital circuit. Thus, under the conventional design approach, <i>timing closure is not certain until after</i> placement.</p>

¹ Note that page numbers do not appear on the original Draft Patent Application, but have been added for convenience. No other changes were made to the Draft Patent Application.

Synopsys Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 3</u></p> <p>If <i>timing closure</i> is not <i>achieved</i> the options the <i>designer</i> has are <i>expensive</i> and unreliable. He may choose to <i>fix the design manually, which is difficult and time consuming, because the automatically optimized network is hard to understand</i>. He may choose to <i>change his HDL specification and repeat the synthesis process</i>. Again <i>timing closure will not be certain until after placement</i>, which means that the entire <i>process</i> needs to be traversed <i>before the designer</i> knows <i>if his HDL changes were successful</i>.</p>	<p><u>'446 PATENT AT 1:48-60</u></p> <p>Failure to <i>achieve timing closure</i> after placement leads to additional <i>expenses</i> and other problems for the <i>designer</i>. To correct for failure to achieve timing closure, the <i>designer</i> has the option of <i>fixing the design manually, which is difficult and time consuming because the automatically optimized digital network is not easy to understand</i>. As a second option, the designer may <i>change the Hardware Description Language (HDL) specification and repeat the design process</i>. However, <i>timing closure will again not be certain until after placement</i>. Thus, the design <i>process</i> must again be repeated <i>before the designer</i> can determine <i>if the HDL specification changes were successful</i> in enabling timing closure.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 3</u></p> <p>A common method of dealing with <i>inaccurate net load estimates</i> is to use <i>net load estimates</i> which are <i>considerably larger than accurate estimates</i>. This <i>causes the sizes of the cells to be considerably larger than necessary</i> but reduces the <i>probability of not meeting the timing constraints after placement</i>. Clearly using cells with sizes <i>which are larger than necessary is wasteful in both silicon area and power consumption</i>. The <i>chips</i> thus synthesized will be <i>larger, cost more to produce and use more electrical power than necessary</i>.</p>	<p><u>'446 PATENT AT 1:61-2:3</u></p> <p>A common method for dealing with <i>inaccurate net load estimates</i> is by <i>estimating the net load</i> at a <i>considerably larger value than typically estimated</i>. Although this method increases the <i>probability of meeting timing constraints after placement</i>, it <i>causes the sizes of the gates to be considerably larger than necessary</i>. Gates which are <i>larger than the necessary size are wasteful in both silicon area and power consumption</i>. This leads to <i>chips</i> which are <i>larger, more expensive to produce, and use more electrical power than necessary</i>.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 3</u></p> <p>A second <i>problem with the conventional approach</i> is that the effect of synthesis decisions is hard to calculate. <i>Performing timing analysis during optimization is very time consuming, and accounts for most of the run time of conventional synthesis systems</i>.</p>	<p><u>'446 PATENT AT 2:4-9</u></p> <p>Another <i>problem with the conventional circuit design approach</i> concerns the timing analysis required <i>during optimization</i> and during placement. The <i>timing analysis performed</i> throughout the conventional circuit design process <i>is very time consuming, and accounts for most of the run time of a conventional circuit design system</i>.</p>

Synopsis Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 4</u></p> <p>In step 105 the placement program will <i>modify the net lengths. Depending on which location was chosen for each cell, the length of each net can be different. As the length differs, the capacitive load of the net changes</i>, and as a result, the delay of the cell driving the net changes. <i>Therefore the delays which were carefully optimized during the logic synthesis, are very different after placement</i>, and the <i>optimization of the network</i> is not very good.</p>	<p><u>'446 PATENT AT 2:12-19</u></p> <p><i>Depending on the location chosen for each gate, each net length may be modified. As each net length is modified, the capacitive load of the net will change. Therefore, the delays, which were carefully optimized during the logic design, are very different in value after cell placement</i>, thereby contributing to poor <i>network optimization</i>.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 4</u></p> <p><i>Much of the progress in the state of the art can be characterized as increased integration.</i> This is represented in figure 1 as various feedback paths, which repeat and alternate steps. The general direction has been towards programs which do structuring, mapping, sizing and placement simultaneously. <i>It has led to increasingly complex software systems which are slow and difficult to design and maintain.</i></p>	<p><u>'446 PATENT AT 2:20-23</u></p> <p>Additionally, <i>much of the progress in the state of the art for digital circuit design can be characterized as increased integration which has led to increasingly complex software systems which are slow, and difficult to design and maintain.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 4</u></p> <p><i>Iterating between placement and sizing</i> has been especially hard to execute because placement programs are not sold by the same design automation software vendors as <i>logic synthesis programs</i>. Also they are not run by the same users: the logic synthesis program is often run by the <i>designer, who also wrote the HDL specification</i>. The <i>placement program is often run by the silicon chip manufacturer, after the design is considered complete.</i></p>	<p><u>'446 PATENT AT 2:24-30</u></p> <p>A further disadvantage with conventional design approaches is in the difficulty of <i>iterating between placement and sizing</i>, since the <i>logic synthesis program</i> is often operated by the <i>logic designer who also wrote the HDL specification</i>, but the <i>placement program</i> is often operated by the <i>silicon chip manufacturer, after the design is complete.</i></p>

Synopsys Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSYS DRAFT PAT. APP. AT 6</u></p> <p>The <i>present invention maintains timing closure</i> after it has been achieved by <i>adjusting the size of the cell during or after placement</i>. The <i>adjustments compensate for the fact that the placement algorithm can assign different net lengths to different nets and that these lengths are difficult to predict before placement</i>.</p>	<p><u>'446 PATENT AT 16:23-29</u></p> <p>According to the <i>present invention, timing closure</i> is <i>maintained</i> after placement occurs of cells 836. To <i>maintain timing closure</i>, the size of a particular gate may be <i>adjusted during or after placement</i>. This <i>adjustment compensates for the fact that placement algorithm may assign different net lengths to different nets, and that these different net lengths are difficult to predict prior to the placement step</i>.</p>
<p><u>SYNOPSYS DRAFT PAT. APP. AT 8</u></p> <p>Wherever possible, the same <i>reference numbers</i> will be used throughout the <i>drawings to refer to the same or like parts</i>.</p>	<p><u>'446 PATENT AT 4:59-63</u></p> <p><i>Referring in detail now to the drawings wherein similar parts or steps of the present invention are identified by like reference numerals</i>, there is seen in FIG. 1 a schematic diagram of a host computer system 100 which is capable of implementing the present invention.</p>
<p><u>SYNOPSYS DRAFT PAT. APP. AT 9</u></p> <p>The cells <i>can be combinational "gates"</i> 207, 208, 209, <i>whose function is represented as an expression in the Boolean algebra, using AND, OR and NOT operators</i>, or the cells <i>can be registers</i> 205, 206.</p>	<p><u>'446 PATENT AT 5:13-17</u></p> <p>The gates <i>can be combinational gates whose function is represented as Boolean expression</i> based on, for example, the <i>operators AND, OR and NOT</i>. The gates <i>can also be registers</i>.</p>
<p><u>SYNOPSYS DRAFT PAT. APP. AT 9</u></p> <p><i>Each cell (e.g., 208) has one or more inputs 212, 213, and a single output 214</i>.</p>	<p><u>'446 PATENT AT 5:18-19</u></p> <p><i>Each gate (e.g., gate j) has one or more input 155 and a single output 160</i>.</p>

Synopsys Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 9</u></p> <p><i>Cells whose inputs are connected to the output of a cell are called the fanin of the latter cell. Cells whose inputs are connected to the output of a cell are called the fanout of the latter cell.</i></p>	<p><u>'446 PATENT AT 5:26-32</u></p> <p><i>Gates whose outputs are connected to the inputs of a gate are collectively called the "fanin" of the latter gate. Thus, the gate k is in the fanin of the gate i. Gates whose inputs are connected to the output of a gate are collectively called the "fanout" of the latter gate. Thus, the gate j is in the fanout of the gate i.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 9</u></p> <p><i>The digital network performs a logic "function" by processing digital binary input data in a number of cycles. The input data is presented to the network on its *primary inputs* 201, 202, and the result of the computation of the network function is presented at the *primary outputs* 203, 204, of the network. The computation of the function takes one or more cycles. During each cycle the gate functions are calculated. The results are stored in the registers for use in the next cycle.</i></p>	<p><u>'446 PATENT AT 5:33-41</u></p> <p><i>The digital circuit 150 performs a logic function by processing digital binary input data in a number of cycles. The input data is presented to the digital circuit 150 at the primary inputs 170, and the result of the computation of the digital circuit function is presented at the primary outputs 175. Typically, the computation of the digital circuit function requires one or more cycles. During each cycle, the gate functions are calculated, and the calculation results are stored in registers for use in the next cycle.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 10</u></p> <p><i>The "arrival time" of the data at a gate is computed by taking the maximum arrival time of its fanin cells each increased by the delay from the input pin to the output pin.</i></p>	<p><u>'446 PATENT AT 9:55-58</u></p> <p><i>(An arrival time of the data at a gate is computed by taking the maximum arrival time of the fanin gates plus the delay measured from the input pin to the output pin of the gate).</i></p>

Synopsys Draft Patent Application	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 11</u></p> <p>The difference between the required time and the arrival time is the <i>*slack*</i>. If the arrival time is smaller than the required time, <i>the timing constraints are met</i>, and the <i>slack is positive</i>. If the arrival time is larger than the required time, the timing constraints are not met, and the slack is negative. The arrival time and required time may be different depending on whether the data is zero (0) or one (1). There also may be multiple arrival times and multiple required times to model a variety of timing constraints. <i>All slacks can be summarized as a single worst slack number</i>, called the <i>*network slack*</i>. <i>Timing closure is achieved if the network slack is non-negative</i>.</p>	<p><u>'446 PATENT AT 13:27-34</u></p> <p>This determination is made by subtracting the delay of the buffer from the "local <i>slack</i>", to give the value of the predicted slack after buffer insertion. <i>Slack is zero or positive if the timing constraints are met</i>. In addition, <i>all slacks in the circuit can be summarized by the "network slack" which is the single "worst" slack number</i>. <i>If the network slack is non-negative</i>, then the <i>timing closure is achieved</i>.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 11</u></p> <p>It is important to note that the dependency on size and load can be captured as the dependency on a single parameter C/S, and <i>the delay D is non-negative and monotonically increasing with C/S</i>.</p>	<p><u>'446 PATENT AT 6:38-43</u></p> <p>The delay D of a gate can be approximated by equation (1):</p> $D=f(C/S) \quad (1)$ <p><i>The delay D is non-negative and increases as the C/S value increases.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 11</u></p> <p><i>The delay may be different for different inputs of the gate and it may be different for the falling and the rising transition.</i></p>	<p><u>'446 PATENT AT 6:58-61</u></p> <p><i>The delay D value may also be different for different inputs of the gate and it may also be different for the falling transition and rising transition of a signal propagating through the gate.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 12</u></p> <p><i>The library analysis will determine a good value for C/S for each cell in the library.</i></p>	<p><u>'446 PATENT AT 6:63-65</u></p> <p><i>The library analysis will determine a "good" value for C/S for each gate in the library based on gain considerations.</i></p>

Synopsis Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 13</u></p> <p>Mostly these optimizations <i>change the structure of the network, and the Boolean functions of the cells, without changing the overall function of the network.</i> The types of <i>optimizations</i> that should be performed are <i>behavioral optimization such as resource sharing, sequential optimizations such as retiming, algebraic optimizations such as kernel extraction and Boolean optimizations such as redundancy removal.</i> These is a large amount of literature on how each of these <i>classes of optimizations</i> can be performed.</p>	<p><u>'446 PATENT AT 9:13-22</u></p> <p>During this step, <i>the structure of the circuit and the Boolean functions of the gates are changed</i> to reduce the total number of connections, <i>without changing the overall function of the circuit.</i> Structural <i>optimizations</i> can include <i>behavioral optimizations (such as resource sharing), sequential optimizations (such as retiming), algebraic optimizations (such as kernel extraction), and Boolean optimizations (such as redundancy removal).</i> The <i>classes of optimizations</i> above are well known to those skilled in the art.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 13</u></p> <p>Following the library independent optimizations, the network <i>is mapped to a library of cells.</i> This means that <i>the logic functions of the cells are implemented with actual cells from the library.</i></p>	<p><u>'446 PATENT AT 9:25-27</u></p> <p>In step 210 (FIG. 4), the circuit <i>is mapped to a library 209 of cells.</i> Thus, <i>the logic functions of the circuit gates are implemented with actual cells from the library 209.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 14</u></p> <p>For example, we can <i>use the "boundary move" transformation to reduce the number of levels in the logic in the mapped network.</i> The boundary move <i>transform</i>, illustrated in fig x, <i>reduces the number of levels by bringing connection x forward.</i></p>	<p><u>'446 PATENT AT 10:45-49</u></p> <p>A local <i>transformation</i> is then <i>used to reduce the number of levels in the logic in the gate chain circuit 550.</i> The result of the <i>transformation</i> is shown as gate chain circuit 550' in FIG. 7B. <i>The number of levels in the logic is reduced by bringing the gate 555 forward.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 14</u></p> <p>To make the change legal <i>it is necessary that gates x, y and z are fanout free. If not, they must be made fanout free</i> by making a copy.</p>	<p><u>'446 PATENT AT 10:59-62</u></p> <p>In order for the transformation shown in FIG. 7B to be valid, <i>it is necessary that gates 555, 560, and 565 are fanout free. If the gates 555, 560, and 565 are not fanout free, then they are made fanout free through copying logic.</i></p>

Synopsis Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 14-15</u></p> <p>In the <i>conventional</i> approach to logic synthesis, <i>copying logic will increase the load on gates x, x, x and therefore increase the delay. To predict if the transformation will improve delay, or hurt delay, it was necessary to run a complete static timing analysis with accurate delay models. If the change actually worsened the delay, then the change would be undone.</i></p>	<p><u>'446 PATENT AT 11:4-13</u></p> <p>Under <i>conventional</i> logic design, <i>copying logic will increase the load on the gates</i> whose outputs are connected to lines 575, 580, 585, and 590. In the example of FIG. 7B, the copying logic 555' <i>increases the load on the gates</i> whose outputs are connected to lines 575 and 580. <i>To predict whether or not the transformation improved delay, it is necessary to run a complete static timing analysis with accurate delay models. If the transformation (from circuit 550 to 550') were actually harmful to delay, then the transformation would have to be undone.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 15</u></p> <p><i>In the constant delay model approach, the effect of this change can be easily predicted. Note that changes in loads do not affect delay. The only change that is affects delay, is the change of the fanin of gate xxx. The delay can easily be predicted by simple addition of gate delays.</i></p>	<p><u>'446 PATENT AT 10:49-58</u></p> <p><i>In the constant delay model approach, the effect of this transformation can be easily predicted. Changes in the gate loads do not affect delay, since delay is maintained as constant while gate size will be adjusted (during or after placement) to compensate for the load change. The only change which affects delay (of the gate chain circuit 550) is the change of the fanin of gate 555. This delay change can be predicted by simple addition of gate delays provided by the fanins connected at lines 590, 575, and 580 (see gate chain circuit 550' in FIG. 7B).</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 15</u></p> <p>The <i>net load</i> consists of the <i>load</i> of the <i>net</i>, which <i>can be estimated using a conventional net load model, plus any other fixed load, such as the load of a primary output.</i></p>	<p><u>'446 PATENT AT 11:26-30</u></p> <p>The parameter <i>w</i> represents the <i>net</i> (wire) <i>load</i> for a given gate <i>i</i> (wherein the <i>net load can be estimated using a conventional net load model</i> such as the above-mentioned fanout-based model) <i>plus any other fixed load such as the load of the primary output</i> of the circuit implementation.</p>

Synopsis Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 15</u></p> <p>In <i>a combinational network</i> this can be achieved by starting <i>at the primary outputs and traversing the network in a leveled order towards the primary inputs</i>.</p>	<p><u>'446 PATENT AT 11:48-52</u></p> <p>If the digital circuit is <i>a combinational network</i> (see, e.g. circuit 150 in FIG. 2), then gate load calculation initiates <i>at the primary outputs 175 and traverses the circuit in a leveled order toward the primary inputs 170</i>.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 15-16</u></p> <p>In <i>a sequential network there may be one or more loops, resulting a cyclic dependency: there is no rightmost cell</i>. In this case the computation <i>can start anywhere in the cycle, and repeats the cycle several times, until the capacitances converge</i> and the error is <i>sufficiently small</i>. It is possible that this iteration will <i>not converge and</i> that the <i>capacitance</i> will <i>increase</i> in every iteration, <i>by progressively larger amounts</i>. This situation is <i>detected</i> by requiring the increment to be smaller than <i>a preset maximum after a fixed number of iterations</i>. The iteration does <i>not converge</i> if the network is <i>an infeasible solution</i>: The current network cannot be <i>expected to work at this speed because its gain is too small</i>. <i>Changes</i> need to be made to the network to <i>increase the gain, which will usually mean increasing the delay</i> of the network as well.</p>	<p><u>'446 PATENT AT 11:53-12:4</u></p> <p>If the digital circuit is <i>a sequential network</i> (see, e.g., circuit 180 of FIG. 3), then <i>there may be one or more loops</i> (e.g., loop 182) which <i>result in a cyclic dependency</i> (i.e., <i>there is no "rightmost" gate</i>). Gate load calculation <i>can start anywhere in the cycle, and calculation in the cycle is performed several times until the load capacitance values converge</i> or have <i>sufficiently small differences</i>. However, a condition may exist when the load <i>capacitance</i> values do <i>not converge and increase by progressively larger amounts</i> every cycle calculation. This increase in load capacitance values can be <i>detected</i> if the calculated load values exceed <i>a preset maximum value after a fixed number of cycle calculations</i>. When the calculated load values do <i>not converge</i>, then the particular circuit 180 has <i>an infeasible solution</i>, which indicates that the digital circuit is <i>not expected to work at the set speed because the circuit gain is too small</i>. <i>Changes</i> are required to <i>increase the circuit gain</i>, and these changes <i>will usually</i> lead to an increase in circuit <i>delay</i>.</p>

Synopsis Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 16</u></p> <p>After the loads have been calculated the <i>size</i> can be calculated <i>by dividing the actual load by the predetermined typical load</i>. The input capacitance can be calculated by multiplying the unit gate input capacitance by the size. The ratio of these numbers is the <i>size</i> of the gate. <i>The size is a scale factor, which</i> can be applied to the area of <i>the gate</i>, to give <i>the area of the sized gate</i>. <i>The area of the network can be estimated as the sum total of the areas of the sized gates, plus the net area as estimated from the total length of all nets.</i></p>	<p><u>'446 PATENT AT 12:5-20</u></p> <p>In the above example, the <i>size</i> S of a gate i is determined <i>by dividing the actual load C_i by the predetermined typical load C/S</i> of the gate i. The size S is a scale factor which is applied to all transistor channel widths of a gate in order to determine the area of the "<i>sized gate</i>". <i>The size S is also a scale factor which</i> is used to scale <i>the gate's</i> output load driving capability and its input pin loads. <i>The area of the sized gate</i> is determined by equation (5).</p> <p>area of sized gate=S*(area of gate) (5)</p> <p><i>The area of the mapped digital circuit can be estimated based on the sum of the total areas of the sized gates plus the net area</i> (which is estimated from the total length of all nets in the circuit).</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 16-17</u></p> <p>We can do this by <i>calculating</i> single parameter per net, called the <i>net weight</i>, which <i>represents the sensitivity of the total area of the network with respect to the load on that net</i>. This <i>net weight</i> can be calculated in a manner that is very similar to the calculation of the pin loads during the area calculation above. Starting at <i>the primary inputs, the net weight of the first (left-most) gate is equal to its area per unit load</i>. <i>The net weights of the other cells can now be calculated</i> with a recurrence relation traversing the network from left to right.</p>	<p><u>'446 PATENT AT 12:22-30</u></p> <p>Thus, the following discussion now turns to the <i>calculation of "net weights."</i> The <i>net weight represents the sensitivity of the total area of a digital circuit with respect to the load of a particular net</i>. As an example, <i>the net weight of a given gate, which is immediately coupled to the primary inputs of a digital circuit, is equal to its area per unit load</i>. Using equation (6), <i>the net weight of the other gates in the digital circuit are then calculated</i> in a leveled order towards the primary outputs of the digital circuit.</p>

Synopsis Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 17</u></p> <p>The <i>buffering</i> algorithm works as follows: First it finds <i>locations in the network where a buffer can be added</i> without increasing the network delay. This is done <i>by subtracting the delay of the buffer from the local slack, to give the predicted slack after buffer insertion. If the predicted slack is larger then the network slack, then a buffer can be inserted without increasing the network delay.</i></p>	<p><u>'446 PATENT AT 13:23-37</u></p> <p>The <i>buffering</i> step of 215 (FIG. 4) is discussed in further detail with reference to FIG. 8. In step 650, <i>locations in the circuit are determined where a buffer can be added</i> so that buffer insertion will still permit timing constraints to be met. This determination is made <i>by subtracting the delay of the buffer from the "local slack", to give the value of the predicted slack after buffer insertion.</i> Slack is zero or positive if the timing constraints are met. In addition, all slacks in the circuit can be summarized by the "network slack" which is the single "worst" slack number. If the network slack is non-negative, then timing closure is achieved. <i>If the predicted slack calculated in step 650 is larger than the network slack, then it is possible to insert a buffer without increasing the circuit delay.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 17-18</u></p> <p>Next we have to calculate the reduction in load of this net, and check that <i>area that is added by adding the buffer does not exceed the area saved by sizing down the source gate.</i> The <i>area added by inserting the buffer</i> is simply <i>the area of the buffer times its size, where the size is determined by the load on the buffer divided by the typical load of the buffer. The area saved by inserting the buffer</i> can be calculated by first <i>calculating the change in load due to the insertion of the buffer: some sinks are removed, the input load of the buffer is added, and the net load estimate may change as a result of the number of fanouts of the net changing.</i></p>	<p><u>'446 PATENT AT 13:37-48</u></p> <p>In step 655, it is determined whether the <i>added area due to buffer insertion does not exceed the area saved by sizing down the source gate.</i> The <i>added area (by inserting the buffer) is equal to the area of the buffer multiplied by the buffer size, wherein the buffer size is determined by the buffer load C divided by the typical load C/S on the buffer. The area saved by sizing down the source gate</i> is determined by first <i>calculating the change in net load due to the buffer insertion.</i> This <i>net load change</i> is due to the following: (1) <i>some sinks (which sink currents) are removed,</i> (2) <i>the input load of the buffer is added, and</i> (3) <i>the number of fanouts of the gate may change.</i></p>

Synopsys Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 18</u></p> <p><i>After the buffer is inserted, the capacitances need to be updated in the fanin cone of the buffer, while the net weights need to be updated in the fanout cone of the buffer.</i></p>	<p><u>'446 PATENT AT 13:53-57</u></p> <p><i>After the buffer has been inserted, then in step 670 the capacitance values need to be updated in the fanin cone of the buffer, while the net weights need to be updated in the fanout cone of the buffer.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 18</u></p> <p>The next step is the process of "<i>Stretching</i>" and "<i>Compressing</i>" the delays of the individual gates to meet the timing constraints. Gates which are on long paths which do not meet the delay constraint are "<i>compressed</i>" until the path does meet the timing constraint. Gates on the short paths which easily meet the timing constraints are "<i>Stretched</i>". Gates with stretched delays require less area for the same load. In this step the delay of the gates is traded against the gain of the gates. When the delay of a gate decreases so does the gain of the gate. It is important that there is enough gain in the network.</p>	<p><u>'446 PATENT AT 14:20-36</u></p> <p>Prior to cell placement, the delays of the individual gates may be stretched or compressed to meet the delay constraints, as shown in step 220 of FIG. 4. As shown in FIG. 9A, by compressing (decreasing) the delay of a given gate, the gate gain decreases. Gates which are on long paths not meeting the delay constraints are compressed (in delay) until the long paths meet the delay constraints. The delay of the gates (or gate) may be decreased as long as the minimum required gain requirements are met. By stretching (increasing) the delay of a given gate, the gate gain increases (see FIG. 9A). Gates on short paths which easily meet the delay constraints are stretched (in delay), since gates with stretched delays require less area for the same load. The delay of the gates (or gate) in a path are stretched to the extent that timing constraints for the digital circuit are still met.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 18-19</u></p> <p><i>For the purpose of stretching and compressing registers can usually be considered to be part of a path which they originate, but not of a path that they terminate.</i></p>	<p><u>'446 PATENT AT 15:48-51</u></p> <p><i>For the purpose of stretching and compressing, registers in the circuit are preferably considered as part of a path from which they originate, but not part of the path from which they terminate.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 19</u></p> <p>The stretching algorithm considers the cells on a path by path basis, processing the path with the smallest slack first.</p>	<p><u>'446 PATENT AT 15:9-10</u></p> <p>The invention operates on a path-by-path basis whereby the most critical path in a digital circuit 750 is evaluated first.</p>

Synopsis Draft Patent Application	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 19</u></p> <p>After a cell <i>has been adjusted</i>, it becomes "<i>locked</i>" and its <i>delay</i> cannot be changed by the <i>stretching</i> algorithm.</p>	<p><u>'446 PATENT AT 15:21-25</u></p> <p>After the gate 754 <i>has been adjusted</i> to meet the Path 2 timing constraints, it becomes "<i>locked</i>," whereby the gate 754 delay will <i>not be</i> adjusted further for the remainder of the compression and <i>stretching</i> step.</p>

1 Chris Scott Graham (State Bar No. 114498)
2 Michael N. Edelman (State Bar No. 180948)

3 **DECHERT LLP**
4 975 Page Mill Road
5 Palo Alto, California 94304
6 Telephone: (650) 813-4800
7 Facsimile: (650) 813-4848

8 Attorneys for Plaintiff SYNOPSYS

ORIGINAL
FILED

SEP 17 2004

RICHARD W. WIEKING
CLERK U.S. DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
SAN JOSE

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN JOSE DIVISION

12 SYNOPSYS, INC., a Delaware corporation

13 Plaintiff,

14 vs.

15 MAGMA DESIGN AUTOMATION, a
16 Delaware corporation,

17 Defendant.

CASE NO. **C04 03923 MEJ**

COMPLAINT FOR PATENT
INFRINGEMENT

DEMAND FOR JURY TRIAL

ADR

E-filing

18
19 Plaintiff SYNOPSYS, INC. ("SYNOPSYS") hereby alleges against Defendant MAGMA
20 DESIGN AUTOMATION ("MAGMA" or "the Defendant") as follows:

21 **JURISDICTION**

22 1. This is an action for patent infringement arising under the patent laws of the United
23 States. This Court has jurisdiction over this action under 28 U.S.C. § 1338(a).

24 **PARTIES**

25 2. SYNOPSYS is a corporation duly organized and existing under the laws of the State
26 of Delaware, with its principal place of business in Mountain View, California.

27 ///

28 ///

3. SYNOPSISYS is informed and believes, and thereon alleges, that MAGMA is a corporation duly organized and existing under the laws of the State of Delaware, with its principal place of business in Santa Clara, California.

VENUE

4. Venue is proper in the Northern District pursuant to 28 U.S.C. § 1391(b) & (c) and 28 U.S.C. § 1400(b).

INTRADISTRICT ASSIGNMENT

5. This is an Intellectual Property Action under this Court's Assignment Plan, and therefore assignment to any division of the Court is proper pursuant to Civil L.R. 3-2(c). SYNOPSISYS believes that assignment to the San Jose division is particularly appropriate given its close proximity to the principal places of business of SYNOPSISYS and MAGMA.

FACTUAL BACKGROUND

6. On or about May 17, 1995, Lukas van Ginneken signed a Proprietary Information and Inventions Agreement (the "Agreement") as a condition to his employment by SYNOPSISYS. Paragraph 3 of this Agreement provides that all rights to any inventions made, conceived, reduced to practice or developed by van Ginneken while employed by SYNOPSISYS are automatically assigned to SYNOPSISYS. A true and correct copy of the Agreement is attached hereto as Exhibit A.

7. While employed by SYNOPSYS, van Ginneken made, conceived and developed inventions pertaining to timing closure methodology, the use of constant delay models in logic synthesis and other aspects of placement and/or synthesis. These inventions were made, conceived and developed by van Ginneken during his employment for SYNOPSYS for the purpose of developing SYNOPSYS' products, and therefore each of these inventions are encompassed by the terms of the Agreement. By operation of law, all right, title and interest to these inventions are automatically assigned to SYNOPSYS under the Agreement.

8. After leaving the employment of SYNOPSIS, van Ginneken co-founded MAGMA. Thereafter, MAGMA submitted patent applications to the Patent and Trademark Office that disclosed inventions that van Ginneken had made, conceived and developed while at SYNOPSIS, and which are owned by SYNOPSIS.

1 9. On April 23, 2002, United States Patent No. 6,378,114 ("the '114 Patent"), entitled
2 "Method for the Physical Placement of an Integrated Circuit Adaptive to Netlist Changes," was
3 issued to SYNOPSYS. van Ginneken is a named inventor on the '114 Patent. A true and correct
4 copy of the '114 Patent is attached to this complaint as Exhibit B and is incorporated by reference
5 herein.

6 10. On September 17, 2002, United States Patent No. 6,453,446 ("the '446 Patent"),
7 entitled "Timing Closure Methodology," was issued to MAGMA. The '446 Patent discloses
8 inventions which were made, conceived and developed by van Ginneken at SYNOPSYS. Pursuant
9 to the terms of the Agreement, SYNOPSYS holds legal and equitable title to the inventions in the
10 '446 Patent. A true and correct copy of the '446 Patent is attached to this complaint as Exhibit C
11 and is incorporated by reference herein.

12 11. On April 20, 2004, United States Patent No. 6,725,438 ("the '438 Patent"), entitled
13 "Timing Closure Methodology," was issued to MAGMA. The '438 Patent contains inventions
14 which were made, conceived and developed by van Ginneken at SYNOPSYS. Pursuant to the terms
15 of the Agreement, SYNOPSYS holds legal and equitable title to the inventions in the '438 Patent. A
16 true and correct copy of the '438 Patent is attached to this complaint as Exhibit D and is incorporated
17 by reference herein.

18 12. Since the issuance of the '114 Patent, '446 Patent, and '438 Patent (collectively ,
19 referred to hereinafter as the "SYNOPSYS PATENTS"), MAGMA has engaged in a wide range of
20 activities to infringe those patents. MAGMA has been involved in making, using, selling,
21 distributing, advertising, marketing and creating source code for products that infringe the
22 SYNOPSYS PATENTS.

23 **FIRST CAUSE OF ACTION**
24 **(PATENT INFRINGEMENT)**

25 13. SYNOPSYS is the owner of the SYNOPSYS PATENTS because, among other
26 reasons, the inventions disclosed in the patents were previously assigned to SYNOPSYS by van
27 Ginneken pursuant to the terms of the Agreement.

28 ///

1 14. MAGMA has been and still is infringing the SYNOPSIS PATENTS in violation of
2 the federal patent laws by making, using, selling, distributing, advertising, marketing and creating
3 source code for products which infringe the SYNOPSIS PATENTS. MAGMA will continue to so
4 infringe unless enjoined by this Court.

5 15. MAGMA has actively induced infringement of, or contributed to the infringement of,
6 the SYNOPSIS PATENTS under the federal patent laws by, among other things, making infringing
7 products and creating source code for infringing products and then selling, distributing, advertising
8 and marketing those infringing products to others, and will continue to do so unless enjoined by this
9 Court.

10 16. MAGMA's infringement of the SYNOPSIS PATENTS in violation of the federal
11 patent laws has been willful and deliberate, and has caused injury to SYNOPSIS.

12 17. MAGMA's infringement in violation of the federal patent laws will continue to injure
13 SYNOPSIS unless enjoined by this Court.

14 WHEREFORE, SYNOPSIS prays for judgment against the Defendant, and requests that this
15 Court impose the following remedies under the federal patent laws:

16 A. Preliminarily and permanently enjoin the Defendant from continued infringement of
17 the SYNOPSIS PATENTS, pursuant to 35 U.S.C. § 283;

18 B. Order the Defendant to account to SYNOPSIS for damages sustained by
19 SYNOPSIS as a result of the Defendant's infringement of the SYNOPSIS PATENTS, with
20 interest, pursuant to 35 U.S.C. § 284;

21 C. Order the Defendant to pay SYNOPSIS a reasonable royalty to compensate for the
22 Defendant's infringement, pursuant to 35 U.S.C. § 284;

23 D. Treble the damages resulting from the Defendant's willful and deliberate
24 infringement, pursuant to 35 U.S.C. § 284;

25 E. Award SYNOPSIS its costs, expenses and reasonable attorneys' fees incurred in
26 bringing and prosecuting this action, pursuant to 35 U.S.C. § 285;

27 ///


28 ///

1 F. Impose a constructive trust for the benefit of SYNOPSIS over any profits, revenues,
2 or other benefits obtained by the Defendant as a result of its infringement of the SYNOPSIS
3 PATENTS; and

4 G. Award SYNOPSIS such further relief that the Court may deem just and proper
5 arising from the Defendant's infringement of the SYNOPSIS PATENTS under the federal patent
6 laws.

7 Dated: September 17, 2004

DECHERT LLP

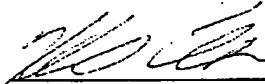
8 
9 Chris Scott Graham
10 Michael Edelman
11 Attorneys for Plaintiff SYNOPSIS
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

DEMAND FOR JURY TRIAL

SYNOPSYS hereby demands trial by jury of all issues.

Dated: September 17, 2004

DECHERT LLP



Chris Scott Graham

Michael Edelman

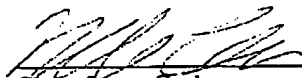
Attorneys for Plaintiff SYNOPSYS

CERTIFICATION OF INTERESTED ENTITIES OR PERSONS

Pursuant to Civil L.R. 3-16, the undersigned certifies that as of this date, other than the named parties, there is no such interest to report.

Dated: September 17, 2004

DECHERT LLP


Chris Scott Graham
Michael Edelman
Attorneys for Plaintiff SYNOPSISYS

PROPRIETARY INFORMATION
AND
INVENTIONS AGREEMENT

The following confirms an agreement between me and Synopsys, Inc. (the "Company"), which is a material part of the consideration for my employment by the Company.

1. I recognize that the Company is engaged in a continuous program of research, development and production respecting its business, present and future, including fields generally related to its business and that the Company possesses and continues to possess information that has been created, discovered, developed or otherwise become known to the Company (including, without limitation, information created, discovered or developed by, or made known to, me during the period of or arising out of my employment by the Company) and/or in which property rights have been assigned, licensed or otherwise conveyed to the Company, which information has commercial value in the business in which the Company is engaged. All of the aforementioned information is hereinafter called "Proprietary Information." By way of illustration, but not limitation, Proprietary Information includes trade secrets, processes, data and know-how, computer software, improvements, inventions, works of authorship, techniques, marketing plans, strategies, forecasts and copyrightable material and customer lists.

2. I understand that my employment creates a relationship of confidence and trust between me and the Company with respect to any information:

(i) applicable to the business of the Company; or

(ii) applicable to the business of any client or customer of the Company, which may be known to me by the Company or by any client or customer of the Company, or learned by me during the period of my employment.

3. In consideration of my employment by the Company and the compensation received by me from the Company from time to time, I hereby agree as follows:

A. All Proprietary Information shall be the sole property of the Company and its assigns, and the Company and its assigns shall be sole owner of all patents, copyrights and other rights in connection therewith. I hereby assign to the Company any rights I may have or acquire in such Proprietary Information. At all times, both during my employment by the Company and after its termination, I will keep in confidence and trust all Proprietary Information, and I will not use or disclose any Proprietary Information or anything relating to it without the written consent of the Company, except as may be necessary in the ordinary course of performing my duties to the Company.

B. All documents, records, apparatus, equipment and other physical property, whether or not pertaining to Proprietary Information, furnished to me by the Company or produced by me or others in connection with my employment shall be and remain the sole property of the Company and shall be returned to the Company immediately as and when requested by the Company. Even if the Company does not so request, I shall return and deliver all such property upon termination of my employment by me or the Company for any reason and I will not take with me any such property or any reproduction of such property upon such termination.

C. I will promptly disclose to the Company, or any persons designated by it, all improvements, inventions, works of authorship, processes, techniques, know-how, formulae,

data, ideas and other information (including, without limitation, my algorithms or software), whether or not patentable, made or conceived or reduced to practice or learned by me, either alone or jointly with others, during the term of my employment (all said improvements, inventions, works of authorship, processes, techniques, know-how, formulae, data, ideas and other information shall be hereinafter collectively called "Inventions").

D. I agree that all Inventions which I make, conceive, reduce to practice or develop (in whole or in part, either alone or jointly with others) during my employment shall be the sole property of the Company to the maximum extent permitted by Section 2870 of the California Labor Code (hereinafter called "Section 2870"), a copy of which is attached hereto as Exhibit B, and to the extent permitted by law, shall be "works made for hire." The Company shall be the sole owner of all patents, copyrights, trade secret rights, rights with respect to other intellectual property or other rights in connection therewith (including, without limitation, such rights in algorithms or software). I hereby assign to the Company any rights I may have or acquire in such Inventions. I agree to perform, during and after my employment, all acts deemed necessary or desirable by the Company to permit and assist it, at the Company's expense, in obtaining, maintaining and enforcing patents, copyrights, trade secret rights, rights with respect to such Inventions and/or other Inventions I have or may at any time assign to the Company in any and all countries. Such acts may include, but are not limited to, execution of documents and assistance or cooperation in legal proceedings. I hereby irrevocably designate and appoint the Company and its duly authorized officers and agents as my agents and attorneys-in-fact to act for and on my behalf and instead of me, to execute and file any applications or related filings and to do all other lawfully permitted acts to further the prosecution, maintenance and enforcement, issuance of patents, copyrights, trade secret rights, rights with respect to mask works or other rights thereon with the same legal force and effect as if executed by me.

E. As a matter of record, I attach hereto a complete list of all inventions or improvements relevant to the subject matter of my employment by the Company which have been made by me or jointly with others prior to my employment with the Company that I desire to remove from the operation of this Agreement, and I covenant that such list is complete. If no such list is attached to this Agreement, I represent that I have no such inventions and improvements at the time of signing this Agreement.

F. During the term of my employment and for one (1) year thereafter, I will not encourage or solicit any employee of the Company to leave the Company for any reason or devote less than all of any such employee's efforts to the affairs of the Company, provided that the foregoing shall not affect any responsibility I may have as an employee if the Company with respect to the bona fide hiring and firing of Company personnel.

G. I represent that my performance of all the terms of this Agreement will not breach any agreement to keep in confidence proprietary information acquired by me in confidence or in trust prior to my employment by the Company. I have not entered into, and I agree I will not enter into, any agreement, either written or oral, in conflict herewith.

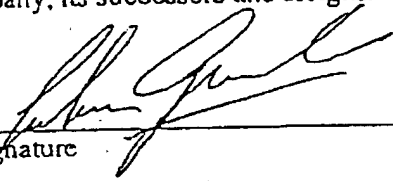
H. I represent that execution of this Agreement, my employment with the Company and my performance of my proposed duties to the Company in the development of its business will not violate any obligations I may have to my former employer.

I. This Agreement does not require assignment of an invention which an employee cannot be obligated to assign under Section 2870. However, I will disclose any Inventions as required by Section 3(c) hereof regardless of whether I believe the Invention is protected by Section 2870, in order to permit the Company to engage in a review process to determine such issues as may arise. Such disclosure shall be received in confidence by the Company.

4. This Agreement shall be effective as of the first day of my employment by the Company: June 26 1995 (date).

5. This Agreement shall be binding upon me, my heirs, executors, assigns and administrators and shall inure to the benefit of the Company, its successors and assigns.

Dated: 5/17/95


Signature

Lukas van Ginneken
(Print or Type Name)

Accepted and agreed to:

Synopsys, Inc.

By  _____

EXHIBIT A

TO

SYNOPSYS, INC. PROPRIETARY INFORMATION
AND INVENTIONS AGREEMENT

1. The following is a complete list of all inventions or improvements relevant to the subject matter of my employment by Synopsys, Inc. (the "Company") that have been made or conceived or first reduced to practice by me alone or jointly with others prior to my employment by the Company that I desire to remove from the operation of the Company's Proprietary Information and Inventions Agreement.

☐ No inventions or improvements

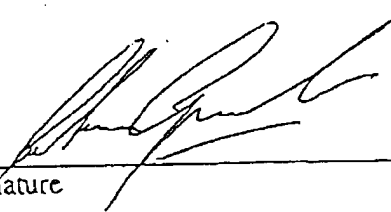
☐ See below

☒ Additional sheets attached

2. I propose to bring to my employment the following materials and documents of a former employer:

☒ No materials or documents

☐ See below



Signature

Lukas van Glnneken

Print or Type Name

EXHIBIT B

TO

SYNOPTIS, INC. PROPRIETARY INFORMATION
AND INVENTIONS AGREEMENT

Section 2870. Application of provision providing that employee shall assign or offer to assign rights in inventions to employer.

(a) Any provision in an employment agreement which provides that an employee shall assign, or offer to assign, any of his or her rights in an invention to her or her employer shall not apply to an invention that the employee entirely on his or her own time without using the employer's equipment, supplies, facilities or trade secret information except for those inventions that either:

(1) Relate at the time of conception or reduction to practice of the invention to the employer's business, or actual or demonstrably anticipated research or development by the employer;

(2) result from any work performed by the employee for the employer.

(b) To the extent a provision in an employment agreement purports to require an employee to assign an invention otherwise excluded from being required to be assigned under subdivision (a), the provision is against the public policy of this state and is unenforceable.

The following is a complete list of all inventions or improvements relevant to the subject matter of my employment by Synopsys, Inc. (the Company) that have been made or conceived or first reduced to practice by me alone or jointly with others prior to my employment by the Company that I desire to remove from the operation of the Company's Proprietary Information and Inventions Agreement.

- [0] "Efficient orthonormality testing for synthesis with pass transistor selectors" by M. R. C. M. Berkelaar and -, accepted at the International Workshop on Logic Synthesis, June 1995.
- [1] "Timing Verification and Optimization for the PowerPC Processor Family", by R.E. Mains, T. A. Mosher, - and R.F. Damiano, in: Proc. Int. Conf. on Computer Design, pp.390-393, Boston, Oct. 10-12, 1994.
- [2] "In the driver's seat of BooleDozer" by D. Brand and R.F. Damiano, -, A. D. Drumm, in: Proc. Int. Conf. on Computer Design, pp. 518-521 Boston, Oct. 10-12, 1994.
- [3] "Grammar-based optimization of synthesis scenarios" by A. Kuehlmann and -, in: Proc. Int. Conf. on Computer Design, pp. 20-25 Boston, Oct. 10-12, 1994.
- [4] "Tuning of logic synthesis scenarios" by - and A. Kuehlmann, Workshop notes of the Int. Workshop on logic synthesis, paper P7c, Tahoe City, May 23-26, 1993.
- [5] "Fanin ordering in multi-slot timing" by -, Proc. Int. Conf. on Computer Design, pp. 44-47, Cambridge, Oct. 11-14, 1992.
- [6] "The complexity of adaptive annealing" by R. H. J. M. Otten and -, Proc. Int. Conf. on Computer Design, pp. 404-407, Cambridge, Sept. 17-19, 1990.
- [7] "Buffer placement in distributed RC-tree networks for minimal Elmore delay" by -, Proc. Int. Symp. on Circuits and Systems, pp. 865-868, New Orleans, May 2-5, 1990.
- [8] "Optimal slicing of plane point placements" by - and R. H. J. M. Otten, Proc. European Design Automation Conf. pp. 322-236, Glasgow, March 12-15, 1990.
- [9] The annealing algorithm by R. H. J. M. Otten and -, ISBN 07923-9022-9, Boston:Kluwer, 1989.
- [10] The predictor-adaptor paradigm - automation of custom layout by flexible design by -, Ph.D. thesis, ISBN 90-9002703-3, Eindhoven, 1989.
- [11] "Doubly folded transistor matrix layout" by - and J. T. J. van Eijndhoven, A. H. C. M. Brouwers, Digest Int. Conf. on Computer Aided Design, Santa Clara, Nov. 7-10, 1988.
- [12] "Stop criteria in simulated annealing" R. H. J. M. Otten and -, Proc.

Int. Conf. on Computer Design, pp.549-552, Port Chester, Oct. 3-5,
1988.

[13] "An inner loop criterion for simulated annealing" by - and R.H.J.M.
Otten, Physics letters A, 130:429-435, 1988.

[14] "Soft Macro Cell generation by two dimensional folding" by - and J.
T. J. van Eijndhoven, P. R. M. van Teeffelen, T. J. Deckers, Proc. Int.
Symp. on Circuits and Systems, pp. 727-730, Espoo, June 1988.

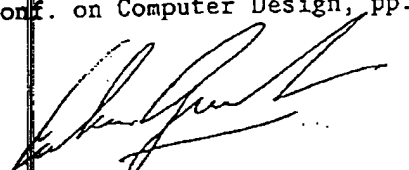
[15] "Gridless routing of general floor plans" by - and J. A. G. Jess, Digest
Int. Conf. on Computer Aided Design, pp. 30-33, Santa Clara Nov. 9-
12, 1987.

[16] "Wire planning for stackable designs", by R. K. Brayton, C. L. Chen,
J. A. G. Jess, R. H. J. M. Otten and -, Proc. Int. Symp. on VLSI tech-
nology, pp.269-273, Taipei, May 13-15, 1987.

[17] "Global wiring for custom layout design" by - and R. H. J. M. Otten,
Proc. Int. Symp. on Circuits and Systems. pp.207-208, Kyoto, June 5-
7, 1985.

[18] "Floor plan design using simulated annealing" by R. H. J. M. Otten
and -, Digest Int. Conf. on Computer Aided Design, pp. 96-98, Santa
Clara, Nov. 1984.

[19] "Stepwise layout refinement" by - and R. H. J. M. Otten, Proc. Int.
Conf. on Computer Design, pp. 30-36, Port Chester, Oct.8-11, 1984.



Lukas van Ginneken

EV 314044781US 28 January 2005

PTO/SB/82 (09-04)

Approved for use through 11/30/2005. OMB 0651-0035

U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

ns are required to respond to a collection of information unless it displays a valid OMB control number.

EV314044781US

**REVOCATION OF POWER OF
ATTORNEY WITH
NEW POWER OF ATTORNEY
AND
CHANGE OF CORRESPONDENCE ADDRESS**

Application Number	10/828,547
Filing Date	19 April 2004
First Named Inventor	van Ginneken
Art Unit	unknown
Examiner Name	unknown
Attorney Docket Number	SYNP 103

I hereby revoke all previous powers of attorney given in the above-identified application.

☐ A Power of Attorney is submitted herewith.

OR

☒ I hereby appoint the practitioners associated with the Customer Number:

36454

☒ Please change the correspondence address for the above-identified application to:

☒ The address associated with
Customer Number:

36454

OR

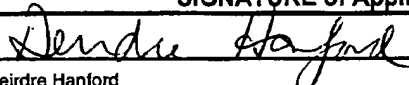
<input type="checkbox"/> Firm or Individual Name				
Address				
City		State		Zip
Country				
Telephone		Fax		

I am the:

☐ Applicant/Inventor.

☒ Assignee of record of the entire interest. See 37 CFR 3.71.
Statement under 37 CFR 3.73(b) is enclosed. (Form PTO/SB/96)

SIGNATURE of Applicant or Assignee of Record

Signature			
Name	Deirdre Hanford		
Date	28 January 2005	Telephone	(650) 584-4201

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required. Submit multiple forms if more than one signature is required, see below.

☐ *Total of _____ forms are submitted.

This collection of information is required by 37 CFR 1.36. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

STATEMENT UNDER 37 CFR 3.73(b)

Applicant/Patent Owner: van Ginneken

Application No./Patent No.: 10/828,547 Filed/Issue Date: 19 April 2004

Entitled: UNKNOWN

SYNOPSYS, INC., a CORPORATION
(Name of Assignee) (Type of Assignee, e.g., corporation, partnership, university, government agency, etc.)

states that it is:

1. ☒ the assignee of the entire right, title, and interest; or
2. ☐ an assignee of less than the entire right, title and interest.
The extent (by percentage) of its ownership interest is _____ %

in the patent application/patent identified above by virtue of either:

A. ☐ An assignment from the inventor(s) of the patent application/patent identified above. The assignment was recorded in the United States Patent and Trademark Office at Reel _____, Frame _____, or for which a copy thereof is attached.

OR

B. ☒ A chain of title from the inventor(s), of the patent application/patent identified above, to the current assignee as shown below:

1. From: Lukas P.P.P. van Ginneken To: Synopsys, Inc.
The document was recorded in the United States Patent and Trademark Office at
Reel _____, Frame _____, or for which a copy thereof is attached.
2. From: _____ To: _____
The document was recorded in the United States Patent and Trademark Office at
Reel _____, Frame _____, or for which a copy thereof is attached.
3. From: _____ To: _____
The document was recorded in the United States Patent and Trademark Office at
Reel _____, Frame _____, or for which a copy thereof is attached.

☒ Additional documents in the chain of title are listed on a supplemental sheet.

☒ Copies of assignments or other documents in the chain of title are attached.

[NOTE: A separate copy (i.e., a true copy of the original assignment document(s)) must be submitted to Assignment Division in accordance with 37 CFR Part 3, if the assignment is to be recorded in the records of the USPTO. See MPEP 302.08]

The undersigned (whose title is supplied below) is authorized to act on behalf of the assignee.

Deirdre Hanford
Signature

28 January 2005
Date

Deirdre Hanford
Printed or Typed Name

(650) 584-4201
Telephone Number

Sr. Vice President
Title

This collection of information is required by 37 CFR 3.73(b). The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 12 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Application No. 10/828,547

SUPPLEMENTAL SHEET TO STATEMENT UNDER 37 CFR 3.73(b)

Attached in support of the chain of title of the referenced application are copies of the following documents:

- 1) a copy of a Proprietary Information and Inventions Agreement attached hereto, signed by the inventor, that assigns to Synopsys, Inc. all inventions made, conceived, reduced to practice, or developed during the inventor's employment with Synopsys, Inc.;
- 2) a copy of a declaration by Robert Damiano, attached hereto, in which Robert Damiano attests that he received a draft patent application from the inventor ("Draft") during the inventor's employment with Synopsys, Inc., including as an attachment the email in which Robert Damiano received the Draft;
- 3) a copy of a chart, attached hereto, highlighting the common language shared between the Draft and the specification of U.S. Patent No. 6,453,446, from which U.S. Patent No. 6,725,438 claims priority as a continuation, from which U.S. Patent Application No. 10/828,547 claims priority as a continuing application, including as an attachment the Draft with page numbers referenced by the chart; and
- 4) a copy of a Complaint for Patent Infringement attached hereto for purposes of disclosure.

PROPRIETARY INFORMATION
AND
INVENTIONS AGREEMENT

The following confirms an agreement between me and Synopsys, Inc. (the "Company"), which is a material part of the consideration for my employment by the Company.

1. I recognize that the Company is engaged in a continuous program of research, development and production respecting its business, present and future, including fields generally related to its business and that the Company possesses and continues to possess information that has been created, discovered, developed or otherwise become known to the Company (including, without limitation, information created, discovered or developed by, or made known to, me during the period of or arising out of my employment by the Company) and/or in which property rights have been assigned, licensed or otherwise conveyed to the Company, which information has commercial value in the business in which the Company is engaged. All of the aforementioned information is hereinafter called "Proprietary Information." By way of illustration, but not limitation, Proprietary Information includes trade secrets, processes, data and know-how, computer software, improvements, inventions, works of authorship, techniques, marketing plans, strategies, forecasts and copyrightable material and customer lists.

2. I understand that my employment creates a relationship of confidence and trust between me and the Company with respect to any information:

(i) applicable to the business of the Company; or

(ii) applicable to the business of any client or customer of the Company, which may be known to me by the Company or by any client or customer of the Company, or learned by me during the period of my employment.

3. In consideration of my employment by the Company and the compensation received by me from the Company from time to time, I hereby agree as follows:

A. All Proprietary Information shall be the sole property of the Company and its assigns, and the Company and its assigns shall be sole owner of all patents, copyrights and other rights in connection therewith. I hereby assign to the Company any rights I may have or acquire in such Proprietary Information. At all times, both during my employment by the Company and after its termination, I will keep in confidence and trust all Proprietary Information, and I will not use or disclose any Proprietary Information or anything relating to it without the written consent of the Company, except as may be necessary in the ordinary course of performing my duties to the Company.

B. All documents, records, apparatus, equipment and other physical property, whether or not pertaining to Proprietary Information, furnished to me by the Company or produced by me or others in connection with my employment shall be and remain the sole property of the Company and shall be returned to the Company immediately as and when requested by the Company. Even if the Company does not so request, I shall return and deliver all such property upon termination of my employment by me or the Company for any reason and I will not take with me any such property or any reproduction of such property upon such termination.

C. I will promptly disclose to the Company, or any persons designated by it, all improvements, inventions, works of authorship, processes, techniques, know-how, formulas,

ATTORNEYS AND
CONSULTANTS ONLY

SY000003

data, ideas and other information (including, without limitation, my algorithms or software), whether or not patentable, made or conceived or reduced to practice or learned by me, either alone or jointly with others, during the term of my employment (all said improvements, inventions, works of authorship, processes, techniques, know-how, formulae, data, ideas and other information shall be hereinafter collectively called "Inventions").

D. I agree that all Inventions which I make, conceive, reduce to practice or develop (in whole or in part, either alone or jointly with others) during my employment shall be the sole property of the Company to the maximum extent permitted by Section 2870 of the California Labor Code (hereinafter called "Section 2870"), a copy of which is attached hereto as Exhibit B, and to the extent permitted by law, shall be "works made for hire." The Company shall be the sole owner of all patents, copyrights, trade secret rights, rights with respect to other intellectual property or other rights in connection therewith (including, without limitation, such rights in algorithms or software). I hereby assign to the Company any rights I may have or acquire in such Inventions. I agree to perform, during and after my employment, all acts deemed necessary or desirable by the Company to permit and assist it, at the Company's expense, in obtaining, maintaining and enforcing patents, copyrights, trade secret rights, rights with respect to such Inventions and/or other Inventions I have or may at any time assign to the Company in any and all countries. Such acts may include, but are not limited to, execution of documents and assistance or cooperation in legal proceedings. I hereby irrevocably designate and appoint the Company and its duly authorized officers and agents as my agents and attorneys-in-fact to act for and on my behalf and instead of me, to execute and file any applications or related filings and to do all other lawfully permitted acts to further the prosecution, maintenance and enforcement, issuance of patents, copyrights, trade secret rights, rights with respect to mask works or other rights thereon with the same legal force and effect as if executed by me.

E. As a matter of record, I attach hereto a complete list of all inventions or improvements relevant to the subject matter of my employment by the Company which have been made by me or jointly with others prior to my employment with the Company that I desire to remove from the operation of this Agreement, and I covenant that such list is complete. If no such list is attached to this Agreement, I represent that I have no such inventions and improvements at the time of signing this Agreement.

F. During the term of my employment and for one (1) year thereafter, I will not encourage or solicit any employee of the Company to leave the Company for any reason or devote less than all of any such employee's efforts to the affairs of the Company, provided that the foregoing shall not affect any responsibility I may have as an employee if the Company with respect to the bona fide hiring and firing of Company personnel.

G. I represent that my performance of all the terms of this Agreement will not breach any agreement to keep in confidence proprietary information acquired by me in confidence or in trust prior to my employment by the Company. I have not entered into, and I agree I will not enter into, any agreement, either written or oral, in conflict herewith.

H. I represent that execution of this Agreement, my employment with the Company and my performance of my proposed duties to the Company in the development of its business will not violate any obligations I may have to my former employer.

I. This Agreement does not require assignment of an invention which an employee cannot be obligated to assign under Section 2870. However, I will disclose any Inventions as required by Section 3(c) hereof regardless of whether I believe the Invention is protected by Section 2870, in order to permit the Company to engage in a review process to determine such issues as may arise. Such disclosure shall be received in confidence by the Company.

SY000004

**ATTORNEYS AND
CONSULTANTS ONLY**

4. This Agreement shall be effective as of the first day of my employment by the Company: June 26 1995 (date).

5. This Agreement shall be binding upon me, my heirs, executors, assigns and administrators and shall inure to the benefit of the Company, its successors and assigns.

Dated: 5/17/95



Signature

Lukas van Ginneken
(Print or Type Name)

Accepted and agreed to:

Synopsys, Inc.

By  _____

ATTORNEYS AND
CONSULTANTS ONLY

SY000005

EXHIBIT A
TO
SYNOPSIS, INC. PROPRIETARY INFORMATION
AND INVENTIONS AGREEMENT

1. The following is a complete list of all inventions or improvements relevant to the subject matter of my employment by Synopsis, Inc. (the "Company") that have been made or conceived or first reduced to practice by me alone or jointly with others prior to my employment by the Company that I desire to remove from the operation of the Company's Proprietary Information and Inventions Agreement.

_____ No inventions or improvements

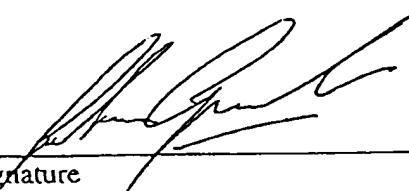
_____ See below

X Additional sheets attached

2. I propose to bring to my employment the following materials and documents of a former employer:

X No materials or documents

_____ See below



Signature

Lukas van Glnneken

Print or Type Name

ATTORNEYS AND
CONSULTANTS ONLY

SY000006

EXHIBIT B

TO

SYNOPSYS, INC. PROPRIETARY INFORMATION
AND INVENTIONS AGREEMENT

Section 2870. Application of provision providing that employee shall assign or offer to assign rights in inventions to employer.

(a) Any provision in an employment agreement which provides that an employee shall assign, or offer to assign, any of his or her rights in an invention to her or her employer shall not apply to an invention that the employee entirely on his or her own time without using the employer's equipment, supplies, facilities or trade secret information except for those inventions that either:

(1) Relate at the time of conception or reduction to practice of the invention to the employer's business, or actual or demonstrably anticipated research or development by the employer;

(2) result from any work performed by the employee for the employer.

(b) To the extent a provision in an employment agreement purports to require an employee to assign an invention otherwise excluded from being required to be assigned under subdivision (a), the provision is against the public policy of this state and is unenforceable.

ATTORNEYS AND
CONSULTANTS ONLY

SY000007

The following is a complete list of all inventions or improvements relevant to the subject matter of my employment by Synopsys, Inc. (the Company) that have been made or conceived or first reduced to practice by me alone or jointly with others prior to my employment by the Company that I desire to remove from the operation of the Company's Proprietary Information and Inventions Agreement.

[0] "Efficient orthonormality testing for synthesis with pass transistor selectors" by M. R. C. M. Berkelaar and -, accepted at the International Workshop on Logic Synthesis, June 1995.

[1] "Timing Verification and Optimization for the PowerPC Processor Family", by R.E. Mains, T. A. Mosher, - and R.F. Damiano, in: Proc. Int. Conf. on Computer Design, pp.390-393, Boston, Oct. 10-12, 1994.

[2] "In the driver's seat of BooleDozer" by D. Brand and R.F. Damiano, -, A. D. Drumm, in: Proc. Int. Conf. on Computer Design, pp. 518-521, Boston, Oct. 10-12, 1994.

[3] "Grammar-based optimization of synthesis scenarios" by A. Kuehlmann and -, in: Proc. Int. Conf. on Computer Design, pp. 20-25 Boston, Oct. 10-12, 1994.

[4] "Tuning of logic synthesis scenarios" by - and A. Kuehlmann, Workshop notes of the Int. Workshop on logic synthesis, paper P7c, Tahoe City, May 23-26, 1993.

[5] "Fanin ordering in multi-slot timing" by -, Proc. Int. Conf. on Computer Design, pp. 44-47, Cambridge, Oct. 11-14, 1992.

[6] "The complexity of adaptive annealing" by R. H. J. M. Otten and -, Proc. Int. Conf. on Computer Design, pp. 404-407, Cambridge, Sept. 17-19, 1990.

[7] "Buffer placement in distributed RC-tree networks for minimal Elmore delay" by -, Proc. Int. Symp. on Circuits and Systems, pp. 865-868, New Orleans, May 2-5, 1990.

[8] "Optimal slicing of plane point placements" by - and R. H. J. M. Otten, Proc. European Design Automation Conf. pp. 322-236, Glasgow, March 12-15, 1990.

[9] The annealing algorithm by R. H. J. M. Otten and -, ISBN 07923-9022-9, Boston:Kluwer, 1989.

[10] The predictor-adaptor paradigm - automation of custom layout by flexible design by -, Ph.D. thesis, ISBN 90-9002703-3, Eindhoven, 1989.

[11] "Doubly folded transistor matrix layout" by - and J. T. J. van Eijndhoven, A. H. C. M. Brouwers, Digest Int. Conf. on Computer Aided Design, Santa Clara, Nov. 7-10, 1988.

[12] "Stop criteria in simulated annealing" R. H. J. M. Otten and -, Proc.

Int. Conf. on Computer Design, pp.549-552, Port Chester, Oct. 3-5, 1988.

[13] "An inner loop criterion for simulated annealing" by - and R.H.J.M. Otten, Physics letters A, 130:429-435, 1988.

[14] "Soft Macro Cell generation by two dimensional folding" by - and J. T. J. van Eijndhoven, P. R. M. van Teeffelen, T. J. Deckers, Proc. Int. Symp. on Circuits and Systems, pp. 727-730, Espoo, June 1988.

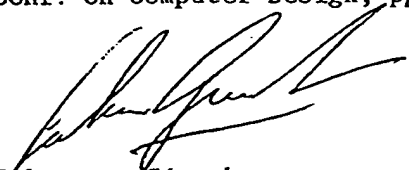
[15] "Gridless routing of general floor plans" by - and J. A. G. Jess, Digest Int. Conf. on Computer Aided Design, pp. 30-33, Santa Clara Nov. 9-12, 1987.

[16] "Wire planning for stackable designs", by R. K. Brayton, C. L. Chen, J. A. G. Jess, R. H. J. M. Otten and -, Proc. Int. Symp. on VLSI technology, pp.269-273, Taipeh, May 13-15, 1987.

[17] "Global wiring for custom layout design" by - and R. H. J. M. Otten, Proc. Int. Symp. on Circuits and Systems. pp.207-208, Kyoto, June 5-7, 1985.

[18] "Floor plan design using simulated annealing" by R. H. J. M. Otten and -, Digest Int. Conf. on Computer Aided Design, pp. 96-98, Santa Clara, Nov. 1984.

[19] "Stepwise layout refinement" by - and R. H. J. M. Otten, Proc. Int. Conf. on Computer Design, pp. 30-36, Port Chester, Oct.8-11, 1984.



Lukas van Ginneken

ATTORNEYS AND
CONSULTANTS ONLY

SY000009

DECLARATION OF ROBERT DAMIANO

I, Robert Damiano, declare as follows:

1. The following statements are based on my personal knowledge. If called upon to testify, I could and would competently testify as to the matters set forth herein.
2. I am an employee of Synopsys, Inc. My present position is Vice-President of the Advanced Technology Group. In September 1996, my position at Synopsys was Director in the Advanced Technology Group. Lukas van Ginneken, who was also employed at Synopsys during that time, worked on one of my projects.
3. On or about September 9, 1996, I received an email from Lukas van Ginneken that included a draft patent application. A true and correct copy of such email is attached hereto as Exhibit A.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of Declarant:

Robert Damiano

Declarant's Signature:

Robert Damiano

Date:

28 January 2005

Exhibit A

[Document Follows]

TO: robertd@synopsys.com
From: Lukas van Ginneken <lukas@synopsys.com>
Subject: patent
Date: 1996-09-09 22:07:54 GMT

SYNOPSYS CONFIDENTIAL

APPLICATION FOR UNITED STATES PATENT

in the name of

LUKAS PAUL PIETER PEPIJN VAN GINNEKEN

of

SYNOPSYS, INC.

for

METHOD FOR ACHIEVING TIMING CLOSURE OF DIGITAL NETWORKS

AND

METHOD FOR AREA OPTIMIZATION OF DIGITAL NETWORKS UNDER TIMING CLOSURE

class 364/489

BACKGROUND OF THE INVENTION

This application relates to a method for achieving timing closure of digital networks consisting of structuring and mapping and a method for area optimization of digital networks using placement and sizing, while maintaining timing closure.

(Prior Art)

Figure 1 shows the conventional approach to digital network synthesis.

Digital network synthesis is a process in which computer programs optimize digital networks. At the beginning of the synthesis process, a human designer 110 specifies a design 111 at a high level of abstraction using a high level design specification language, such as Verilog or VHDL.

In step 101 the abstract network specification of the design is transformed into an unmapped digital network representation in memory by a computer program, such as HDL Compiler, available from Synopsys Inc. of Mountain View, California.

In step 102 Logic synthesis algorithms optimize the network by changing the structure of the network without changing the function of the network.

In step 103, logic network synthesis algorithms map the abstract network representation to cells from the library.

Step 104 optimizes the size of the cells. The size of the cell, together with the load of the cell determines the delay of the cell. The sizing algorithm adjusts the sizes of the cells. Changing the sizes of the cells affects delay and area, and the sizing algorithm manipulates the cell sizes so as to minimize delay and area.

Steps 102, 103 and 104 are performed by a computer program, such as "Design Compiler" TM available from Synopsys Inc., of Mountain View, California.

Step 105 determines the placement of the cells on the chip. Placement algorithms attempt to keep the length of the nets short, as longer nets need more area on the chip and the increased net load of longer nets will make the network slower. The network remains unchanged during the placement.

Finally step 106 determines the exact routing of the nets on the chip. Steps 105 and 106 are done by a computer program, such as "Cell Ensemble" TM available from Cadence Inc. of San Jose, California.

(Problems with prior art)

The major problem with the conventional approach is that the net length and hence the cell delay is not known until after placement. Before placement, net length must be estimated. This is usually done with an estimation function or table which gives the load of a net based on its fanout. Experience has shown that it is very difficult to estimate the length of the nets accurately. Essentially net length behaves as a random variable.

The result is unpleasant surprises after placement step 105. Some nets turn out to be longer than expected, and because of the longer delays the timing constraints are not met. Timing closure is not certain until after step 105.

If timing closure is not achieved the options the designer has are expensive and unreliable. He may choose to fix the design manually, which is difficult and time consuming, because the automatically optimized network is hard to understand. He may choose to change his HDL specification and repeat the synthesis process. Again timing closure will not be certain until after placement, which means that the entire process needs to be traversed before the designer knows if his HDL changes were successful.

A common method of dealing with inaccurate net load estimates is to use net load estimates which are considerably larger than accurate estimates. This causes the sizes of the cells to be considerably larger than necessary but reduces the probability of not meeting the timing constraints after placement. Clearly using cells with sizes which are larger than necessary is wasteful in both silicon area and power consumption. The chips thus synthesized will be larger, cost more to produce and use more electrical power than necessary.

A second problem with the conventional approach is that the effect

of synthesis decisions is hard to calculate. Performing timing analysis during optimization is very time consuming, and accounts for most of the run time of conventional synthesis systems.

In step 103 it is difficult to take decisions based on delay without knowing load and size of the cells as well.

In step 104, changing the size of a cell affects the loads of the fanin cells, and thus the delay of the fanin cells. In more complex delay models, which take into account the transition time of the signals, also the delay of the fanout cells is affected. Usually the size parameter cannot have any arbitrary value. Because the library of cells has been designed before the network synthesis started, only a few sizes are available. 3 or 4 sizes per cell is common. This makes it harder to find a good solution.

In step 105 the placement program will modify the net lengths. Depending on which location was chosen for each cell, the length of each net can be different. As the length differs, the capacitive load of the net changes, and as a result, the delay of the cell driving the net changes. Therefore the delays which were carefully optimized during the logic synthesis, are very different after placement, and the optimization of the network is not very good.

Much of the progress in the state of the art can be characterized as increased integration. This is represented in figure 1 as various feedback paths, which repeat and alternate steps. The general direction has been towards programs which do structuring, mapping, sizing and placement simultaneously. It has lead to increasingly complex software systems which are slow and difficult to design and maintain.

Iterating between placement and sizing has been especially hard to execute because placement programs are not sold by the same design automation software vendors as logic synthesis programs. Also they are not run by the same users: the logic synthesis program is often run by the designer, who also wrote the HDL specification. The placement program is often run by the silicon chip manufacturer, after the design is considered complete.

SUMMARY OF THE INVENTION

(objectives)

It is an object of the present invention to achieve timing closure as quickly as possible in the synthesis process. This will give the human designer early warning if the design is infeasible, because it is over constrained.

The invention achieves this object by

- speeding up the algorithms in the synthesis process
- performing the feasibility check early in the synthesis process, that is, before placement, instead of after placement.
- maintaining feasibility throughout the remainder of the synthesis process, so that it can be guaranteed to succeed and can be executed automatically.

The present invention overcomes the problems of the conventional approach by not choosing a size for a cell at all. Rather than choosing a default size, as conventional methods do, we choose a delay and let the size implicitly be whatever it needs to be to meet that delay.

In the conventional method of optimization, the structure, mapping, size and placement are chosen to optimize delay and area. In our formulation of the problem, we choose the structure, mapping, delay and placement, to optimize size and area. In our formulation, size only affects the area, so area only remains as an optimization goal.

The present invention speeds up the programs by simplifying or eliminating timing analysis. While optimizing network delay in order to achieve timing closure, the delays are constant, which will speed up delay calculation. Also, since changes to the network do not change the delay of the cells, the amount of recalculation is drastically reduced. While optimizing area after timing closure has been achieved, timing analysis is not needed, as the delay of the cells does not change. The one step where timing analysis using complex delay models is necessary is in the stretching step. Here too, library design rules are taken into account.

The present invention maintains timing closure after it has been achieved by adjusting the size of the cell during or after placement. The adjustments compensate for the fact that the placement algorithm can assign different net lengths to different nets and that these lengths are difficult to predict before placement.

(language of the main claims)

In accordance with the purpose of this invention, as embodied and broadly described herein, the invention is a method for the structuring and mapping of an unmapped digital network comprising the following steps:

- a) structuring of the digital network, using network slack as an optimization goal, where network slack is calculated assuming that the delay of the cells of the network is constant with respect to load.
- b) mapping of the cells of the network, using network slack as an optimization goal, where network slack is calculated assuming that the delay of the cells of the network is constant with respect to load.
- c) estimation of the area of the network based on net load

In further accordance with the purpose of this invention, as embodied and broadly described herein, the invention is a method for the placement and sizing of cells of a mapped digital network, the method comprising the steps of:

- d) Choosing a target delay for each cell.
- e) Computing the network slack using the target delays.
- f) Placement of the cells of the network.
- g) Sizing of the cells of the network such that the network meets the network slack as computed by step b).

In further accordance with the purpose of this invention, as embodied and broadly described herein, the invention is a method for the placement of the cells of a digital network, the method comprising the steps, of:

- f1) The calculation of net weights that reflect the change of network area due to sizing as a function of net length.
- f2) Placement of the cells of the network where the weighted network net length is used as a placement objective, the weighted network net length being the sum of the weighted net lengths of all nets, each net length being

multiplied by a weight,

Objects and advantages of the invention will be set forth in part in the description which follows and in part will be obvious from the description or may be learned by practice of the invention. The objects and advantages of the invention will be realized and attained by means of the elements and combinations particularly pointed out in the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute part of this specification, illustrate several embodiments of the invention and, together with the description, serve to explain the principles of the invention.

Fig. 1 is a flow chart showing the flow of the conventional method.

Fig. 2 is a flow chart showing the flow according to the present invention.

Fig. 3 is a block diagram of a computer.

Fig. 4 is a schematic diagram of a digital network.

Fig. 5 is a schematic diagram of the electronic and the physical implementations of a cell.

Fig. 6 is a timing diagram illustrating the concept of slack and other timing concepts.

Fig. 7 is a graph showing the relationship between the delay of a cell, the size of a cell and the load of a cell.

Fig. 10 is a

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Reference will now be made in detail to the preferred embodiments of the invention, examples of which are illustrated in the accompanying drawings. Wherever possible, the same reference numbers will be used throughout the drawings to refer to the same or like parts.

(what insight led to this invention?)

(Software patent application)

The preferred embodiment of the present invention is as one or more computer programs. The digital network, its elements and attributes, exist during the process of the method only as data structures in the memory of the computer. Methods in prior art are known to persons of ordinary skill in the art to convert the design data in memory eventually to an actual physical implementation of the network.

(Computer system)

Figure 1 is a block diagram of a computer system 100 in accordance with

the present invention. Computer system 100 includes a central processing unit 101, bus 102, memory 103, input device 104 and output device 105. It will be understood by a person of ordinary skill in the art that computer system 100 can also include numerous elements not shown in the figure for the sake of clarity, such as disk drives, tape drives, mice, printers, network connections, additional CPUs, etc. Memory 103 contains a program 107, which embodies the invention, and a data structure representation of the network 106.

(Network terminology)

Figure 2 is a schematic diagram of a digital network. Digital *network* 200 is composed of a plurality of *cells* 205, 206, 207, 208, 209, connected by a plurality of *nets*. Each cell (e.g. 208) has one or more inputs 212, 213, and a single output 214. Each net is connected to one output and one or more inputs. The cells can be combinational *gates* 207, 208, 209, whose function is represented as an expression in the Boolean algebra, using AND, OR and NOT operators, or the cells can be registers 205, 206. All feed back loops 210, 211 in the network contain at least one register. Cells whose outputs are connected to the inputs of a cell are called the fanin of the latter cell. Cells whose inputs are connected to the output of a cell are called the fanout of the latter cell.

(Computation)

The digital network performs a logic *function* by processing digital binary input data in a number of cycles. The input data is presented to the network on its *primary inputs* 201, 202, and the result of the computation of the network function is presented at the *primary outputs* 203, 204, of the network. The computation of the function takes one or more cycles. During each cycle the gate functions are calculated. The results are stored in the registers for use in the next cycle. The data flow is from inputs to outputs and will be assumed to be pictured as going from left to right throughout this text.

(Mapping)

Each cell can be *mapped* to a *book* in which case a electronic realization in terms of transistors has been chosen, and physical attributes such as delay are known. If a cell is not mapped to a book it is *unmapped*, in which case no electronic realization has been selected, and the function of the cell is only known in abstract terms, such as Boolean algebraic expressions. A *library* of books is designed in advance, before synthesis starts. These books are of generic types, and can be used to build arbitrary designs.

(Size)

The cell has a delay, an area and its input pins have an input pin capacitance. The *size* of the cell is a multiplier which is applied to equally to all transistor channel widths in the electronic circuit of the cell. Thus size of a cell is a scale factor which is used to scale its output load driving capability (see below), its area, and its input pin loads.

(Timing constraints)

Each primary input or primary output has an associated delay,

called the *input delay*, respectively *output delay*, which represents delays external to the network. The network needs a certain amount of time to perform one cycle, called the *cycle delay*. Together the cycle delay, the input delays and the output delays form the timing constraints of the network. Meeting the timing constraints is called *timing closure* and it is a major objective of the synthesis process.

(Timing)

The delay of a path is measured as the sum of the gate delays over said path from begin point to end point. The cycle delay is the maximum of all path delays. Primary outputs and register inputs are timing path end points. Primary inputs and register outputs are timing path begin points.

(Slack)

The *arrival time* of the data at a gate is computed by taking the maximum arrival time of its fanin cells each increased by the delay from the input pin to the output pin. The arrival times can be computed by traversing the network from left to right, that is, from timing begin points to timing end points. Similarly the *required time* of the data at a gate can be computed by traversing the network from end points to begin points. The required time is the minimum required time of its fanout cells, each reduced by the delay from the input to the output pin of that fanout cell.

The difference between the required time and the arrival time is the *slack*. If the arrival time is smaller then the required time, the timing constraints are met, and the slack is positive. If the arrival time is larger then the required time, the timing constraints are not met, and the slack is negative. The arrival time and required time may be different depending on whether the data is zero (0) or one (1). There also may be multiple arrival times and multiple required times to model a variety of timing constraints. All slacks can be summarized as a single worst slack number, called the *network slack*. Timing closure is achieved if the network slack is non-negative.

(Delay model)

The delay D of a gate depends on many factors, among them its function, its size S and the capacitive load C of the gate. The delay may be different for different inputs of the gate and it may be different for the falling and the rising transition. It is important to note that the dependency on size and load can be captured as the dependency on a single parameter C/S , and the delay D is non-negative and monotonically increasing with C/S .

$$\begin{aligned} C &> 0 \\ S &> 0 \\ D &= f(C/S) \\ f(C/S) &> 0 \\ f'(C/S) &> 0 \end{aligned}$$

Figure 4 illustrates the relationship between the three variables, size S , delay D and capacitive load C . Each of the three planes shows the relationship between two variables, while the third variable is constant, (not necessarily zero).

(Detailed description)

The preferred embodiment of the invention is a software program that can be stored in the memory of a computer, and can be executed by the central processing unit of the computer so that the computer performs the method described herein.

The software program consists of many parts or subprograms which together perform the method described in this invention. The essence of the invention is that logic synthesis is done in a size independent way, and that sizes are determined after placement, and that we guarantee that the delay numbers before placement can be met by sizing after placement.

The preferred embodiment consists of three parts: (See fig...) to wit

- a) The logic synthesis program
- b) The placement program
- c) The sizing program

- analyze the library
- read logic
- library independent optimization.

The first step is to analyze the library that will be used for logic synthesis. The library contains the cells that will be used to implement the logic function. Contrary to the standard method of performing logic synthesis, we will assume that each cell can be sized by a continuous, positive real variable S , which increases both the load driving capability of the cell and the area linearly. In other words, the area of a cell is $S \cdot A$ and the delay of a cell is $D = f(C/S)$.

The library analysis will determine a good value for C/S for each cell in the library. Using this value, it determines a constant delay for each gate.

(choice of C/S)

Since the library analysis is not dependent on the actual network being synthesized, library analysis can be performed before beginning the synthesis process. We will now continue to describe the actual automatic synthesis process, beginning at the with reading the design. The design is expressed in a high level design specification language, for example VHDL or Verilog, and is syntactically parsed and transformed into a logic network representation by a computer program, such as HDL Compiler, available from Synopsys Inc. of Mountain View, California.

(library independent optimization)

Initially the network is library independent and library independent optimizations are performed. Mostly these optimizations change the structure of the network, and the Boolean functions of the cells, without changing the overall function of the network. The types of optimizations that should be performed are behavioral optimizations, such as resource sharing, sequential optimizations such as retiming, algebraic optimizations such as kernel extraction and Boolean optimizations such as redundancy removal. There is a large amount of literature on how each of these classes of optimizations can be performed.

(mapping for delay)

Following the library independent optimisations the network is mapped to a library of cells. This means that the logic functions of the cells are implemented with actual cells from the library. During this process the A large body of literature exists already on the subject of mapping digital networks. The preferred embodiment would use a previously published algorithm such as

(post mapping optimizations)

Due to restrictions in run time, it is impractical to explore the entire design space during the mapping algorithm. Necessarily, the mapping algorithm has to ignore many possible solutions because either they are unlikely candidates or they are very similar (but not identical) to other, considered solutions. In addition, in the constant delay approach, it is easy to evaluate the impact on timing of synthesis decisions, but it is much harder to evaluate the impact on the total network area. Therefore the mapping algorithm necessarily cannot accurately optimize area.

(pin swapping)

An example of candidate mappings which are not explored during mapping because they are too similar to other mappings follows here: Often gates have several pins which are functionally interchangeable. For instance for a 4 input NAND gate, there are $4! = 24$ possible permutations for the 4 input pins. Usually these pins are not equally fast, because of an inherent asymmetry of the electronic circuit. Because the differences are small, it is not worthwhile to consider all of these different of different permutations during mapping. It is more efficient to pick one arbitrarily and to select the best permutation of the inputs after mapping.

(structuring - boundary move)

Using constant delay it is considerably easier to predict the effect of a change to the network than with the conventional delay models. This can be used to do timing optimization by means of restructuring after technology mapping has been done. For example, we can use the "boundary move" transformation to reduce the number of levels in the logic in the mapped network. The boundary move transform, illustrated in fig x, reduces the number of levels by bringing connection x forward. To make the change legal it is necessary that gates x, y and z are fanout free. If not, they must be made fanout free by making a copy. In the conventional approach to logic synthesis, copying logic will increase the load on gates x, x, x and therefore increase the delay. To predict if the transformation will improve delay, or hurt delay, it was necessary to run a complete static timing analysis with accurate delay models. If the change actually worsened the delay, then the change would be undone.

In the constant delay model approach, the effect of this change can be easily predicted. Note that changes in loads do not affect delay. The only change that affects delay, is the change of the fanin of gate xxx. The delay can easily be predicted by simple addition of gate delays.

(area estimation)

To perform area optimization it is necessary to calculate the sizes of the cells. The sizes can be calculated in a straightforward manner from the loads. The loads are calculated by adding the net load and the pin load. The net load consists of the load of the net, which can be estimated using a conventional net load model, plus any other fixed load, such as the load of a primary output. The pin load is not fixed, that is, the load of an input pin depends on the size of the cell. This creates a dependency: To calculate the load of a cell, we need to calculate the size of its fanout cells. Therefore the algorithm starts calculating as far downstream as possible, and traverse the network in a direction opposite to the flow of data. In a combinational network this can be achieved by starting at the primary outputs and traversing the network in a levelized order towards the primary inputs. In a sequential network there may be one or more loops, resulting a cyclic dependency: there is no rightmost cell. In this case the computation can start anywhere in the cycle, and repeats the cycle several times, until the capacitances converge and the error is sufficiently small.

It is possible that this iteration will not converge and that the capacitance will increase in every iteration, by progressively larger amounts. This situation is detected by requiring the increment to be smaller than a preset maximum after a fixed number of iterations. The iteration does not converge if the network is an infeasible solution: The current network cannot be expected to work at this speed because its gain is too small. Changes need to be made to the network to increase the gain, which will usually mean increasing the delay of the network as well.

After the loads have been calculated the size can be calculated by dividing the actual load by the predetermined typical load. The input capacitance can be calculated by multiplying the unit gate input capacitance by the size. The ratio of these numbers is the size of the gate. The size is a scale factor, which can be applied to the area of the gate, to give the area of the sized gate. The area of the network can be estimated as the sum total of the areas of the sized gates, plus the net area as estimated from the total length of all nets.

(net weights)

Various algorithms, such as buffering and placement optimize the network by manipulating the loads in the network. Placement optimizes the net length, which directly related to the net load, and buffering reduces the load on a cell by adding extra delay. These algorithms can benefit from a more efficient calculation of the effect that changing the load of a cell has on sizing. We can do this by calculating single parameter per net, called the net weight, which represents the sensitivity of the total area of the network with respect to the load on that net.

This net weight can be calculated in a manner that is very similar to the calculation of the pin loads during the area calculation above. Starting at the primary inputs, the net weight of the first (left-most) gate is equal to its area per unit load. The net weights of the other cells can now be calculated with a recurrence relation traversing the network from left to right.....

to the calculation of the loads. Starting at the primary inputs the network to 0. Set the net load on the net in question to 1. Perform the iteration and calculate the area as described in the previous section.

Since all calculations are linear, the effects simultaneous changes in loads of several nets can be superimposed, that is, added together.

(buffering)

The next step in the synthesis process is the buffering step. The buffering algorithm adds buffers to the network guided by the timing analysis and the area analysis. In the constant delay model, the buffers have a fixed delay, and thus the impact of inserting a buffer is easily determined by subtracting the delay of the buffer from the slack. Thus the effect of adding a buffer on delay is always negative: a buffer always adds delay, never reduces delay. The main effect of adding a buffer is to save area because the source gate can be smaller because the it's load is smaller. The effect on area and gain can be determined by area analysis. Net weights.....

The buffering algorithm works as follows: First it finds locations in the network where a buffer can be added without increasing the network delay. This is done by subtracting the delay of the buffer from the local slack, to give the predicted slack after buffer insertion. If the predicted slack is larger than the network slack, then a buffer can be inserted without increasing the network delay. Next we have to calculate the reduction in load of this net, and check that area that is added by adding the buffer does not exceed the area saved by sizing down the source gate. The area added by inserting the buffer is simply the area of the buffer times its size, where the size is determined by the load on the buffer divided by the typical load of the buffer. The area saved by inserting the buffer can be calculated by first calculating the change in load due to the insertion of the buffer: some sinks are removed, the input load of the buffer is added, and the net load estimate may change as a result of the number of fanouts of the net changing. Using the net weight, we can estimate the impact on the network area. If the impact is positive (reduced area) the buffer is inserted. After the buffer is inserted, the capacitances need to be updated in the fanin cone of the buffer, while the net weights need to be updated in the fanout cone of the buffer.

(stretching)

The next step is the process of "Stretching" and "Compressing" the delays of the individual gates to meet the timing constraints. Gates which are on long paths which do not meet the delay constraint are "compressed" until the path does meet the timing constraint. Gates on the short paths which easily meet the timing constraints are "stretched". Gates with stretched delays require less area for the same load. In this step the delay of the gates is traded against the gain of the gates. When the delay of a gate decreases so does the gain of the gate. It is important that there is enough gain in the network.

Note that

this can be done entirely independent of the sizes of the gates. Sizes of gates are not determined until much later in the process.

The stretching algorithm has two phases. In the first phase it will compress the delays of cells on long paths to meet timing constraints. In the second phase it will stretch the delays of the

gates on short paths to save area. For the purpose of stretching and compressing registers can usually be considered to be part of a path which they originate, but not of a path that they terminate. The stretching algorithm considers the cells on a path by path basis, processing the path with the smallest slack first. The delay of each cell on the path is adjusted by an amount which is equal to the slack divided by the number of stages on the path. After a cell has been adjusted, it becomes "locked" and its delay cannot be changed by the stretching algorithm. Stages which are locked are not counted when calculating the adjustments. For the stretching phase, the algorithm continues as above. In this phase the delays of the cells are increased, not decreased. The path that we work on is not the worst path, but it is the worst path with a slack greater than 0. (All other paths now have a slack of 0).

(Rule based stretching)

(Incremental rule based)

(placement)

In our process a conventional placement method is augmented to optimize the area of the placed network. All placement methods known work by gradual refinement of the placement. Periodically, during the placement process, we recalculate the estimated net lengths, using the most recent, accurate placement information. From the net lengths, it calculates the sizes of the cells in the network. (See area estimation). The updated sizes can then be used for further placement and for more accurate net length calculations.

Placement primarily manipulates the lengths of the nets. Using the net weights, the area of the network can efficiently be estimated.

(final or discrete sizing)

Other embodiments will be apparent to those skilled in the art from consideration of the specification and practice of the invention disclosed herein. It is intended that the specification and examples be considered as exemplary only, with a true scope of the invention being indicated by the following claims.

WHAT IS CLAIMED IS:

(CONSTANT DELAY SYNTHESIS)

1. A method for the structuring and mapping of an unmapped digital network comprising the following steps:
 - a) structuring of the digital network, using network slack as an optimization goal, where network slack is calculated assuming that the delay of the cells of the network is constant with respect to load.
 - b) mapping of the cells of the network, using network slack as an optimization goal, where network slack is calculated assuming that the delay of the cells of the network is constant with respect to load.
 - c) estimation of the area of the network based on net load

(typical load)

3. The method of claim 1, the delay of a book is used for the delay of each cell, the delay of the book being chosen by choosing a load size ratio C/S for each book, which is independent of the network.

4. The method of claim 3, where the delay of a book is also determined by the choice of the input transition time, which is independent of the network.

(continuous buffering assumption)

5. The method of claim 3,

where a parameter C/S for each book

is chosen to have the largest

possible value such that a long chain of cells of identical books each cell in the chain having identical value of parameter C/S, said chain cannot have simultaneously improved delay and improved gain by adding a buffer at some point to the same chain, even when the parameter C/S is chosen optimally after adding the buffer.

(buffering)

6. The method of claim 1, with the additional step of buffer insertion before step c), the buffers being inserted on paths with positive slack, as determined by subtracting the delay of the buffer from the slack of the path.

(more buffering)

7. The method of claim 6, where the buffer is inserted if area is saved.

8. The method of claim 7, where the area savings are estimated using net weights which reflect the change of network area due to sizing as a function of net length.

9. The method of claim 8, where the calculation of the net weights

is performed by starting at the primary inputs and traversing the network in the direction of the data flow, while calculating the net weight of a cell by summing over the fanin of the cell the product of the net weight of the fanin cell divided by the gain plus the cells area/load sensitivity.

as $W_i = \sum_j W_j / g_{ij}$

(stretching)

10. The method of claim 1, where step a) is comprising of:

a1) choosing a delay for each book

a2) the delay of the cell assuming the delay of its book

a2) adjusting the delay of each cell based on the slack

11. The method of claim 10, where in step a2) the delay of each cell is adjusted equally among the stages which have the same slack.

12. The method of claim 10, where in step a2) the delay is adjusted on each path, such that the slack of each path becomes 0.

(globally optimal mapping)

13. The method of claim 1, where in step b) the mapping is performed in two steps:

b1) a traversal of the network from

primary inputs and registers from left to right, while choosing at each cell the fastest matching books from all available matching books, using the constant delays of the books and the fastest arrival times of the fanins

of the matching book.

b2) a traversal of the network from right to left, while choosing at each cell the fastest matching book from the candidates selected during the previous traversal.

(Area estimation)

14. The method of claim 1, where in step c) consists of the following steps:

- c1) estimation of the net length based on the number of fanout cells.
- c2) estimation of the capacitive load of the cells using the net length
- c3) calculation of the sizes from the capacitive load.
- c4) calculation of the network area by summation of the product of the book area times the cell size.

(sizing algorithm)

15. The method of claim 14, where step c2) is performed by starting at the primary outputs and traversing the network in the direction opposite to the data flow, while calculating the load of a cell, by summing over the fanout of the cell, the product of the load of the fanout cell divided by the gain plus the net load of the cell.

(iterate till convergence)

16. The method of claim 15, where the network has loops, the traversal is started at primary outputs and arbitrary cells in the loop, and the calculation is repeated until convergence.

17. The method of claim 16, where the traversal is started at primary outputs and registers.

(Area optimization)

18. The method of claim 1, where steps a) and b) also use network area as an optimization goal, in addition to network delay, the network area being estimated as in step c).

(retiming)

19. The method of claim 1, where the structuring step is preceded by a retiming step, where registers are moved in the network, while preserving the function of the network, and where change to the network do not affect the delay of the individual cells.

(I could produce a bunch more of these sort of claims)

(TIMING CLOSURE)

20. A method for the placement and sizing of cells of a mapped digital network, the method comprising the steps of:

- d) Choosing a target delay for each cell.
- e) Computing the network slack using the target delays.
- f) Placement of the cells of the network.
- g) Sizing of the cells of the network such that the network meets the network slack as computed by step b).

(sizing for better placement)

21. The method of claim 20, where step f) is performed in gradual steps, each step being followed by a sizing step g)

(partitioning)

22. The method of claim 21, where step f) is performed by repeated partitioning steps, partitioning the cells in the network into two or more groups, each group being assigned to an subdivision of the plane, alternating the partitioning steps with sizing steps, essentially similar to step g)

(iterative improvement)

23. The method of claim 21, where step f) is performed by choosing an arbitrary initial location in the two dimensional plane for each cell, the placement being optimized by repeatedly changing the location of one or two cells at a time, while performing a sizing step, essentially similar to step g) after each location change.

(sizing)

24. The method of claim 5, where step g) is consists of the following steps:
g1) calculation of the net length based on the available placement information
g2) calculation of the capacitive load of the cells using the net length
g3) calculation of the sizes from the capacitive load.

(sizing algorithm)

25. The method of claim 24, where step g2) is performed by starting at the primary outputs and traversing the network in the direction opposite to the data flow, while calculating the load of a cell, by summing over the fanout of the cell, the product of the load of the fanout cell divided by the gain plus the net load of the cell.

(iterate till convergence)

26. The method of claim 10, where the network has loops, the traversal is started at primary outputs and arbitrary cells in the loop, and the calculation is repeated until convergence.

27. The method of claim 26, where the traversal is started at primary outputs and registers.

(cell generation)

28. The method of claim 20, where following step g) the layout of the cells is generated automatically to yield the exact transistor sizes calculated by step g)

(discrete sizing)

29. The method of claim 24, where step g3) consists of selecting the most suitable size from a limited set of available sizes.

(Weighted Placement)

30. A method for the placement of the cells of a digital network, the method comprising the steps, of:

f1) The calculation of net weights that reflect the change of network area due to sizing

as a function of net length.

f2) Placement of the cells of the network

where the weighted network net length is used as

a placement objective, the weighted network net length being the sum

of the weighted net lengths of all nets, each net length being multiplied by a weight,

(Calculation of net weights)

31. The method of claim 30, where step f1)

is performed by starting at the primary inputs and traversing the network in the direction of the data flow, while calculating the net weight of a cell by summing over the fanin of the cell the product of the net weight of the fanin cell divided by the gain plus the cells area/load sensitivity.
as $W_i = \sum_j W_j / g_{ij}$

(iterate till convergence)

32. The method of claim 31, where the network has loops, the traversal is started at primary inputs and arbitrary cells in the loop, and the calculation is repeated until convergence.

33. The method of claim 32, where the traversal is started at primary inputs and registers.

(Weighted Placement for power)

34. A method for the placement of the cells of a digital network, the method comprising the steps, of:

f1) The calculation of net weights that reflect the change of network power due to sizing as a function of net length.

f2) Placement of the cells of the network where the weighted network net length is used as a placement objective, the weighted network net length being the sum of the weighted net lengths of all nets, each net length being multiplied by a weight,

(Calculation of net weights)

35. The method of claim 30, where step f1) is performed by starting at the primary inputs and traversing the network in the direction of the data flow, while calculating the net weight of a cell by summing over the fanin of the cell the product of the net weight of the fanin cell divided by the gain plus the cells power/load sensitivity.
as $W_i = \sum_j W_j / g_{ij}$

(iterate till convergence)

36. The method of claim 31, where the network has loops, the traversal is started at primary inputs and arbitrary cells in the loop, and the calculation is repeated until convergence.

37. The method of claim 32, where the traversal is started at primary inputs and registers.

38. The method of claim 35, where the power/load sensitivity of a cell is calculated as the product of the switching frequency at that cell times the square of the voltage times the capacitance, where capacitance is calculated as the sum of the net load plus the internal capacitance of the cell, scaled with its size.

ABSTRACT

A method for the design of digital networks consisting of a plurality of cells. The invention uses constant delays during logic synthesis and sizes the cells after placement so as to meet the cycle delay predicted before synthesis. The method chooses a constant delay before logic synthesis and guarantees that it can maintain this delay after placement by means of sizing. Thus it overcomes the unpredictable effects of placement on the cycle delay. The invention also describes methods for choosing the sizes of the gates and a method for inserting buffers.

Synopsys Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 2</u></p> <p>The major problem with the <i>conventional approach</i> is that the net length and hence the cell delay is not known <i>until after placement</i>.</p>	<p><u>'446 PATENT AT 1:46-47</u></p> <p>Thus, under the <i>conventional</i> design <i>approach</i>, timing closure is not certain <i>until after placement</i>.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 2-3</u></p> <p>Before placement, <i>net length</i> must be <i>estimated</i>. This is usually done with <i>an estimation function or table which gives the load of a net based on its fanout</i>. Experience has shown that it is very difficult to <i>estimate</i> the length of the nets <i>accurately</i>.</p>	<p><u>'446 PATENT AT 1:37-40</u></p> <p>While <i>net lengths</i> have been <i>estimated</i> prior to placement by use of <i>an estimation function or table which gives the load value of a net based on</i> the number of <i>fanout</i> gates, this <i>estimation</i> function is usually <i>inaccurate</i>.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 3</u></p> <p>The result is unpleasant surprises <i>after placement</i> step 105. <i>Some nets turn out to be longer than expected</i>, and because of the <i>longer delays</i>, the <i>timing constraints</i> are not met. <i>Timing closure is not certain until after</i> step 105.</p>	<p><u>'446 PATENT AT 1:41:46</u></p> <p>This difficulty in accurately predicting net lengths leads to unpredictable delay effects <i>after cell placement</i> occurs. For example, <i>some nets turn out to be longer in length than expected</i>. These longer nets cause <i>longer delays</i> which prevent satisfaction of <i>timing constraints</i> in the digital circuit. Thus, under the conventional design approach, <i>timing closure is not certain until after</i> placement.</p>

¹ Note that page numbers do not appear on the original Draft Patent Application, but have been added for convenience. No other changes were made to the Draft Patent Application.

Synopsis Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 3</u></p> <p>If <i>timing closure</i> is not <i>achieved</i> the options the <i>designer</i> has are <i>expensive</i> and unreliable. He may choose to <i>fix the design manually, which is difficult and time consuming, because the automatically optimized network is hard to understand.</i> He may choose to <i>change his HDL specification and repeat the synthesis process.</i> Again <i>timing closure will not be certain until after placement,</i> which means that the entire <i>process</i> needs to be traversed <i>before the designer</i> knows <i>if his HDL changes were successful.</i></p>	<p><u>'446 PATENT AT 1:48-60</u></p> <p>Failure to <i>achieve timing closure</i> after placement leads to additional <i>expenses</i> and other problems for the <i>designer.</i> To correct for failure to achieve timing closure, the <i>designer</i> has the option of <i>fixing the design manually, which is difficult and time consuming because the automatically optimized digital network is not easy to understand.</i> As a second option, the designer may <i>change the Hardware Description Language (HDL) specification and repeat the design process.</i> However, <i>timing closure will again not be certain until after placement.</i> Thus, the design <i>process</i> must again be repeated <i>before the designer</i> can determine <i>if the HDL specification changes were successful</i> in enabling timing closure.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 3</u></p> <p>A common method of dealing with <i>inaccurate net load estimates</i> is to use <i>net load estimates</i> which are <i>considerably larger than accurate estimates.</i> This <i>causes the sizes of the cells to be considerably larger than necessary</i> but reduces the <i>probability of not meeting the timing constraints after placement.</i> Clearly using cells with <i>sizes which are larger than necessary is wasteful in both silicon area and power consumption.</i> The <i>chips</i> thus synthesized will be <i>larger, cost more to produce and use more electrical power than necessary.</i></p>	<p><u>'446 PATENT AT 1:61-2:3</u></p> <p>A common method for dealing with <i>inaccurate net load estimates</i> is by <i>estimating the net load</i> at a <i>considerably larger value than typically estimated.</i> Although this method increases the <i>probability of meeting timing constraints after placement,</i> it <i>causes the sizes of the gates to be considerably larger than necessary.</i> Gates which are <i>larger than the necessary size are wasteful in both silicon area and power consumption.</i> This leads to <i>chips</i> which are <i>larger, more expensive to produce, and use more electrical power than necessary.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 3</u></p> <p>A second <i>problem with the conventional approach</i> is that the effect of synthesis decisions is hard to calculate. <i>Performing timing analysis during optimization is very time consuming, and accounts for most of the run time of conventional synthesis systems.</i></p>	<p><u>'446 PATENT AT 2:4-9</u></p> <p>Another <i>problem with the conventional circuit design approach</i> concerns the timing analysis required <i>during optimization</i> and during placement. The <i>timing analysis performed</i> throughout the conventional circuit design process <i>is very time consuming, and accounts for most of the run time of a conventional circuit design system.</i></p>

Synopsis Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 4</u></p> <p>In step 105 the placement program will <i>modify the net lengths. Depending on which location was chosen for each cell, the length of each net can be different. As the length differs, the capacitive load of the net changes</i>, and as a result, the delay of the cell driving the net changes. <i>Therefore the delays which were carefully optimized during the logic synthesis, are very different after placement</i>, and the <i>optimization of the network</i> is not very good.</p>	<p><u>'446 PATENT AT 2:12-19</u></p> <p><i>Depending on the location chosen for each gate, each net length may be modified. As each net length is modified, the capacitive load of the net will change. Therefore, the delays, which were carefully optimized during the logic design, are very different in value after cell placement</i>, thereby contributing to poor <i>network optimization</i>.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 4</u></p> <p><i>Much of the progress in the state of the art can be characterized as increased integration.</i> This is represented in figure 1 as various feedback paths, which repeat and alternate steps. The general direction has been towards programs which do structuring, mapping, sizing and placement simultaneously. <i>It has led to increasingly complex software systems which are slow and difficult to design and maintain.</i></p>	<p><u>'446 PATENT AT 2:20-23</u></p> <p>Additionally, <i>much of the progress in the state of the art for digital circuit design can be characterized as increased integration which has led to increasingly complex software systems which are slow, and difficult to design and maintain.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 4</u></p> <p><i>Iterating between placement and sizing</i> has been especially hard to execute because placement programs are not sold by the same design automation software vendors as <i>logic synthesis programs</i>. Also they are not run by the same users: the logic synthesis program is often run by the <i>designer, who also wrote the HDL specification</i>. The <i>placement program is often run by the silicon chip manufacturer, after the design is considered complete.</i></p>	<p><u>'446 PATENT AT 2:24-30</u></p> <p>A further disadvantage with conventional design approaches is in the difficulty of <i>iterating between placement and sizing</i>, since the <i>logic synthesis program</i> is often operated by the <i>logic designer who also wrote the HDL specification</i>, but the <i>placement program is often operated by the silicon chip manufacturer, after the design is complete.</i></p>

Synopsys Draft Patent Application	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSYS DRAFT PAT. APP. AT 6</u></p> <p>The <i>present invention maintains timing closure</i> after it has been achieved by <i>adjusting the size of the cell during or after placement</i>. The <i>adjustments compensate for the fact that the placement algorithm can assign different net lengths to different nets and that these lengths are difficult to predict before placement</i>.</p>	<p><u>'446 PATENT AT 16:23-29</u></p> <p>According to the <i>present invention, timing closure is maintained</i> after placement occurs of cells 836. To <i>maintain timing closure</i>, the size of a particular gate may be <i>adjusted during or after placement</i>. This <i>adjustment compensates for the fact that placement algorithm may assign different net lengths to different nets, and that these different net lengths are difficult to predict prior to the placement step</i>.</p>
<p><u>SYNOPSYS DRAFT PAT. APP. AT 8</u></p> <p>Wherever possible, the same <i>reference numbers</i> will be used throughout the <i>drawings to refer to the same or like parts</i>.</p>	<p><u>'446 PATENT AT 4:59-63</u></p> <p><i>Referring in detail now to the drawings wherein similar parts or steps of the present invention are identified by like reference numerals</i>, there is seen in FIG. 1 a schematic diagram of a host computer system 100 which is capable of implementing the present invention.</p>
<p><u>SYNOPSYS DRAFT PAT. APP. AT 9</u></p> <p>The cells <i>can be combinational "gates"</i> 207, 208, 209, <i>whose function is represented as an expression in the Boolean algebra, using AND, OR and NOT operators</i>, or the cells <i>can be registers</i> 205, 206.</p>	<p><u>'446 PATENT AT 5:13-17</u></p> <p>The gates <i>can be combinational gates whose function is represented as Boolean expression</i> based on, for example, the <i>operators AND, OR and NOT</i>. The gates <i>can also be registers</i>.</p>
<p><u>SYNOPSYS DRAFT PAT. APP. AT 9</u></p> <p><i>Each cell (e.g., 208) has one or more inputs 212, 213, and a single output 214.</i></p>	<p><u>'446 PATENT AT 5:18-19</u></p> <p><i>Each gate (e.g., gate j) has one or more input 155 and a single output 160.</i></p>

Synopsys Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 9</u></p> <p><i>Cells whose inputs are connected to the output of a cell are called the fanin of the latter cell. Cells whose inputs are connected to the output of a cell are called the fanout of the latter cell.</i></p>	<p><u>'446 PATENT AT 5:26-32</u></p> <p><i>Gates whose outputs are connected to the inputs of a gate are collectively called the "fanin" of the latter gate. Thus, the gate k is in the fanin of the gate i. Gates whose inputs are connected to the output of a gate are collectively called the "fanout" of the latter gate. Thus, the gate j is in the fanout of the gate i.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 9</u></p> <p><i>The digital network performs a logic "function" by processing digital binary input data in a number of cycles. The input data is presented to the network on its *primary inputs* 201, 202, and the result of the computation of the network function is presented at the *primary outputs* 203, 204, of the network. The computation of the function takes one or more cycles. During each cycle the gate functions are calculated. The results are stored in the registers for use in the next cycle.</i></p>	<p><u>'446 PATENT AT 5:33-41</u></p> <p><i>The digital circuit 150 performs a logic function by processing digital binary input data in a number of cycles. The input data is presented to the digital circuit 150 at the primary inputs 170, and the result of the computation of the digital circuit function is presented at the primary outputs 175. Typically, the computation of the digital circuit function requires one or more cycles. During each cycle, the gate functions are calculated, and the calculation results are stored in registers for use in the next cycle.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 10</u></p> <p><i>The "arrival time" of the data at a gate is computed by taking the maximum arrival time of its fanin cells each increased by the delay from the input pin to the output pin.</i></p>	<p><u>'446 PATENT AT 9:55-58</u></p> <p><i>(An arrival time of the data at a gate is computed by taking the maximum arrival time of the fanin gates plus the delay measured from the input pin to the output pin of the gate).</i></p>

Synopsys Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 11</u></p> <p>The difference between the required time and the arrival time is the <i>*slack*</i>. If the arrival time is smaller than the required time, <i>the timing constraints are met</i>, and the <i>slack is positive</i>. If the arrival time is larger than the required time, the timing constraints are not met, and the slack is negative. The arrival time and required time may be different depending on whether the data is zero (0) or one (1). There also may be multiple arrival times and multiple required times to model a variety of timing constraints. <i>All slacks can be summarized as a single worst slack number</i>, called <i>the *network slack*</i>. <i>Timing closure is achieved if the network slack is non-negative</i>.</p>	<p><u>'446 PATENT AT 13:27-34</u></p> <p>This determination is made by subtracting the delay of the buffer from the "local <i>slack</i>", to give the value of the predicted slack after buffer insertion. <i>Slack is zero or positive if the timing constraints are met</i>. In addition, <i>all slacks in the circuit can be summarized by the "network slack" which is the single "worst" slack number</i>. <i>If the network slack is non-negative, then the timing closure is achieved</i>.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 11</u></p> <p>It is important to note that the dependency on size and load can be captured as the dependency on a single parameter C/S, and <i>the delay D is non-negative and monotonically increasing with C/S</i>.</p>	<p><u>'446 PATENT AT 6:38-43</u></p> <p>The delay D of a gate can be approximated by equation (1):</p> $D=f(C/S) \quad (1)$ <p><i>The delay D is non-negative and increases as the C/S value increases.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 11</u></p> <p><i>The delay may be different for different inputs of the gate and it may be different for the falling and the rising transition.</i></p>	<p><u>'446 PATENT AT 6:58-61</u></p> <p><i>The delay D value may also be different for different inputs of the gate and it may also be different for the falling transition and rising transition of a signal propagating through the gate.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 12</u></p> <p><i>The library analysis will determine a good value for C/S for each cell in the library.</i></p>	<p><u>'446 PATENT AT 6:63-65</u></p> <p><i>The library analysis will determine a "good" value for C/S for each gate in the library based on gain considerations.</i></p>

Synopsis Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 13</u></p> <p>Mostly these optimizations <i>change the structure of the network, and the Boolean functions of the cells, without changing the overall function of the network.</i> The types of <i>optimizations</i> that should be performed are <i>behavioral optimization such as resource sharing, sequential optimizations such as retiming, algebraic optimizations such as kernel extraction and Boolean optimizations such as redundancy removal.</i> These is a large amount of literature on how each of these <i>classes of optimizations</i> can be performed.</p>	<p><u>'446 PATENT AT 9:13-22</u></p> <p>During this step, <i>the structure of the circuit and the Boolean functions of the gates are changed</i> to reduce the total number of connections, <i>without changing the overall function of the circuit.</i> Structural <i>optimizations</i> can include <i>behavioral optimizations (such as resource sharing), sequential optimizations (such as retiming), algebraic optimizations (such as kernel extraction), and Boolean optimizations (such as redundancy removal).</i> The <i>classes of optimizations</i> above are well known to those skilled in the art.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 13</u></p> <p>Following the library independent optimizations, the network <i>is mapped to a library of cells.</i> This means that <i>the logic functions of the cells are implemented with actual cells from the library.</i></p>	<p><u>'446 PATENT AT 9:25-27</u></p> <p>In step 210 (FIG. 4), the circuit <i>is mapped to a library 209 of cells.</i> Thus, <i>the logic functions of the circuit gates are implemented with actual cells from the library 209.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 14</u></p> <p>For example, we can <i>use</i> the “boundary move” <i>transformation to reduce the number of levels in the logic in the mapped network.</i> The boundary move <i>transform</i>, illustrated in fig x, <i>reduces the number of levels by bringing connection x forward.</i></p>	<p><u>'446 PATENT AT 10:45-49</u></p> <p>A local <i>transformation</i> is then <i>used to reduce the number of levels in the logic in the gate chain circuit 550.</i> The result of the <i>transformation</i> is shown as gate chain circuit 550' in FIG. 7B. <i>The number of levels in the logic is reduced by bringing the gate 555 forward.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 14</u></p> <p>To make the change legal <i>it is necessary that gates x, y and z are fanout free. If not, they must be made fanout free</i> by making a copy.</p>	<p><u>'446 PATENT AT 10:59-62</u></p> <p>In order for the transformation shown in FIG. 7B to be valid, <i>it is necessary that gates 555, 560, and 565 are fanout free. If the gates 555, 560, and 565 are not fanout free, then they are made fanout free through copying logic.</i></p>

Synopsis Draft Patent Application	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 14-15</u></p> <p>In the <i>conventional</i> approach to logic synthesis, <i>copying logic will increase the load on gates x, x, x</i> and therefore increase the delay. <i>To predict if the transformation will improve delay, or hurt delay, it was necessary to run a complete static timing analysis with accurate delay models. If the change actually worsened the delay, then the change would be undone.</i></p>	<p><u>'446 PATENT AT 11:4-13</u></p> <p>Under <i>conventional</i> logic design, <i>copying logic will increase the load on the gates</i> whose outputs are connected to lines 575, 580, 585, and 590. In the example of FIG. 7B, the copying logic 555' <i>increases the load on the gates</i> whose outputs are connected to lines 575 and 580. <i>To predict whether or not the transformation improved delay, it is necessary to run a complete static timing analysis with accurate delay models. If the transformation (from circuit 550 to 550') were actually harmful to delay, then the transformation would have to be undone.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 15</u></p> <p><i>In the constant delay model approach, the effect of this change can be easily predicted. Note that changes in loads do not affect delay. The only change that is affects delay, is the change of the fanin of gate xxx. The delay can easily be predicted by simple addition of gate delays.</i></p>	<p><u>'446 PATENT AT 10:49-58</u></p> <p><i>In the constant delay model approach, the effect of this transformation can be easily predicted. Changes in the gate loads do not affect delay, since delay is maintained as constant while gate size will be adjusted (during or after placement) to compensate for the load change. The only change which affects delay (of the gate chain circuit 550) is the change of the fanin of gate 555. This delay change can be predicted by simple addition of gate delays provided by the fanins connected at lines 590, 575, and 580 (see gate chain circuit 550' in FIG. 7B).</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 15</u></p> <p>The <i>net load</i> consists of the <i>load</i> of the <i>net</i>, which <i>can be estimated using a conventional net load model, plus any other fixed load, such as the load of a primary output.</i></p>	<p><u>'446 PATENT AT 11:26-30</u></p> <p>The parameter <i>w</i> represents the <i>net</i> (wire) <i>load</i> for a given gate <i>i</i> (wherein the <i>net load can be estimated using a conventional net load model</i> such as the above-mentioned fanout-based model) <i>plus any other fixed load such as the load of the primary output</i> of the circuit implementation.</p>

Synopsis Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 15</u></p> <p>In <i>a combinational network</i> this can be achieved by starting <i>at the primary outputs and traversing the network in a leveled order towards the primary inputs</i>.</p>	<p><u>'446 PATENT AT 11:48-52</u></p> <p>If the digital circuit is <i>a combinational network</i> (see, e.g. circuit 150 in FIG. 2), then gate load calculation initiates <i>at the primary outputs 175 and traverses the circuit in a leveled order toward the primary inputs 170</i>.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 15-16</u></p> <p>In <i>a sequential network there may be one or more loops, resulting a cyclic dependency: there is no rightmost cell</i>. In this case the computation <i>can start anywhere in the cycle, and repeats the cycle several times, until the capacitances converge</i> and the error is <i>sufficiently small</i>. It is possible that this iteration will <i>not converge</i> and that the <i>capacitance will increase</i> in every iteration, <i>by progressively larger amounts</i>. This situation is <i>detected</i> by requiring the increment to be smaller than <i>a preset maximum after a fixed number of iterations</i>. The iteration does <i>not converge</i> if the network is <i>an infeasible solution</i>: The current network <i>cannot be expected to work at this speed because its gain is too small</i>. <i>Changes</i> need to be made to the network to <i>increase the gain</i>, which <i>will usually mean increasing the delay</i> of the network as well.</p>	<p><u>'446 PATENT AT 11:53-12:4</u></p> <p>If the digital circuit is <i>a sequential network</i> (see, e.g., circuit 180 of FIG. 3), then <i>there may be one or more loops</i> (e.g., loop 182) which <i>result in a cyclic dependency</i> (i.e., <i>there is no "rightmost" gate</i>). Gate load calculation <i>can start anywhere in the cycle, and calculation in the cycle is performed several times until the load capacitance values converge</i> or have <i>sufficiently small</i> differences. However, a condition may exist when the load <i>capacitance</i> values do <i>not converge and increase by progressively larger amounts</i> every cycle calculation. This increase in load capacitance values can be <i>detected</i> if the calculated load values exceed <i>a preset maximum value after a fixed number of cycle calculations</i>. When the calculated load values do <i>not converge</i>, then the particular circuit 180 has <i>an infeasible solution</i>, which indicates that the digital circuit is <i>not expected to work at the set speed because the circuit gain is too small</i>. <i>Changes</i> are required to <i>increase the circuit gain</i>, and these changes <i>will usually</i> lead to an increase in circuit <i>delay</i>.</p>

Synopsis Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 16</u></p> <p>After the loads have been calculated the <i>size</i> can be calculated <i>by dividing the actual load by the predetermined typical load</i>. The input capacitance can be calculated by multiplying the unit gate input capacitance by the size. The ratio of these numbers is the <i>size of the gate</i>. <i>The size is a scale factor, which</i> can be applied to the area of <i>the gate</i>, to give <i>the area of the sized gate</i>. <i>The area of the network can be estimated as the sum total of the areas of the sized gates, plus the net area as estimated from the total length of all nets.</i></p>	<p><u>'446 PATENT AT 12:5-20</u></p> <p>In the above example, the <i>size S</i> of a gate <i>i</i> is determined <i>by dividing the actual load C_i by the predetermined typical load C/S of the gate i</i>. The size <i>S</i> is a scale factor which is applied to all transistor channel widths of a gate in order to determine the area of the "<i>sized gate</i>". <i>The size S is also a scale factor which</i> is used to scale <i>the gate's</i> output load driving capability and its input pin loads. <i>The area of the sized gate</i> is determined by equation (5).</p> <p>area of sized gate = $S \times (\text{area of gate})$ (5)</p> <p><i>The area of the mapped digital circuit can be estimated based on the sum of the total areas of the sized gates plus the net area (which is estimated from the total length of all nets in the circuit).</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 16-17</u></p> <p>We can do this by <i>calculating</i> single parameter per net, called the <i>net weight</i>, which <i>represents the sensitivity of the total area of the network with respect to the load on that net</i>. This <i>net weight</i> can be calculated in a manner that is very similar to the calculation of the pin loads during the area calculation above. Starting at <i>the primary inputs, the net weight of the first (left-most) gate is equal to its area per unit load</i>. <i>The net weights of the other cells can now be calculated</i> with a recurrence relation traversing the network from left to right.</p>	<p><u>'446 PATENT AT 12:22-30</u></p> <p>Thus, the following discussion now turns to the <i>calculation of "net weights."</i> The <i>net weight represents the sensitivity of the total area of a digital circuit with respect to the load of a particular net</i>. As an example, <i>the net weight of a given gate, which is immediately coupled to the primary inputs of a digital circuit, is equal to its area per unit load</i>. Using equation (6), <i>the net weight of the other gates in the digital circuit are then calculated</i> in a leveled order towards the primary outputs of the digital circuit.</p>

Synopsis Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 17</u></p> <p>The <i>buffering</i> algorithm works as follows: First it finds <i>locations in the network where a buffer can be added</i> without increasing the network delay. This is done <i>by subtracting the delay of the buffer from the local slack, to give the predicted slack after buffer insertion. If the predicted slack is larger then the network slack, then a buffer can be inserted without increasing the network delay.</i></p>	<p><u>'446 PATENT AT 13:23-37</u></p> <p>The <i>buffering</i> step of 215 (FIG. 4) is discussed in further detail with reference to FIG. 8. In step 650, <i>locations in the circuit are determined where a buffer can be added</i> so that buffer insertion will still permit timing constraints to be met. This determination is made <i>by subtracting the delay of the buffer from the "local slack", to give the value of the predicted slack after buffer insertion.</i> Slack is zero or positive if the timing constraints are met. In addition, all slacks in the circuit can be summarized by the "network slack" which is the single "worst" slack number. If the network slack is non-negative, then timing closure is achieved. <i>If the predicted slack calculated in step 650 is larger than the network slack, then it is possible to insert a buffer without increasing the circuit delay.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 17-18</u></p> <p>Next we have to calculate the reduction in load of this net, and check that <i>area that is added by adding the buffer does not exceed the area saved by sizing down the source gate.</i> The <i>area added by inserting the buffer</i> is simply <i>the area of the buffer times its size, where the size is determined by the load on the buffer divided by the typical load of the buffer.</i> The <i>area saved by inserting the buffer</i> can be calculated by first <i>calculating the change in load due to the insertion of the buffer: some sinks are removed, the input load of the buffer is added, and the net load estimate may change as a result of the number of fanouts of the net changing.</i></p>	<p><u>'446 PATENT AT 13:37-48</u></p> <p>In step 655, it is determined whether the <i>added area</i> due to <i>buffer</i> insertion <i>does not exceed the area saved by sizing down the source gate.</i> The <i>added area (by inserting the buffer)</i> is equal to <i>the area of the buffer multiplied by the buffer size, wherein the buffer size is determined by the buffer load C divided by the typical load C/S on the buffer.</i> The <i>area saved by sizing down the source gate</i> is determined by first <i>calculating the change in net load due to the buffer insertion.</i> This <i>net load change</i> is due to the following: (1) <i>some sinks (which sink currents) are removed,</i> (2) <i>the input load of the buffer is added, and</i> (3) <i>the number of fanouts of the gate may change.</i></p>

Synopsys Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 18</u></p> <p><i>After the buffer is inserted, the capacitances need to be updated in the fanin cone of the buffer, while the net weights need to be updated in the fanout cone of the buffer.</i></p>	<p><u>'446 PATENT AT 13:53-57</u></p> <p><i>After the buffer has been inserted, then in step 670 the capacitance values need to be updated in the fanin cone of the buffer, while the net weights need to be updated in the fanout cone of the buffer.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 18</u></p> <p>The next step is the process of "<i>Stretching</i>" and "<i>Compressing</i>" the delays of the individual gates to meet the timing constraints. Gates which are on long paths which do not meet the delay constraint are "<i>compressed</i>" until the path does meet the timing constraint. Gates on the short paths which easily meet the timing constraints are "<i>Stretched</i>". Gates with stretched delays require less area for the same load. In this step the delay of the gates is traded against the gain of the gates. When the delay of a gate decreases so does the gain of the gate. It is important that there is enough gain in the network.</p>	<p><u>'446 PATENT AT 14:20-36</u></p> <p>Prior to cell placement, the delays of the individual gates may be stretched or compressed to meet the delay constraints, as shown in step 220 of FIG. 4. As shown in FIG. 9A, by compressing (decreasing) the delay of a given gate, the gate gain decreases. Gates which are on long paths not meeting the delay constraints are compressed (in delay) until the long paths meet the delay constraints. The delay of the gates (or gate) may be decreased as long as the minimum required gain requirements are met. By stretching (increasing) the delay of a given gate, the gate gain increases (see FIG. 9A). Gates on short paths which easily meet the delay constraints are stretched (in delay), since gates with stretched delays require less area for the same load. The delay of the gates (or gate) in a path are stretched to the extent that timing constraints for the digital circuit are still met.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 18-19</u></p> <p><i>For the purpose of stretching and compressing registers can usually be considered to be part of a path which they originate, but not of a path that they terminate.</i></p>	<p><u>'446 PATENT AT 15:48-51</u></p> <p><i>For the purpose of stretching and compressing, registers in the circuit are preferably considered as part of a path from which they originate, but not part of the path from which they terminate.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 19</u></p> <p>The stretching algorithm considers the cells on a path by path basis, processing the path with the smallest slack first.</p>	<p><u>'446 PATENT AT 15:9-10</u></p> <p>The invention operates on a path-by-path basis whereby the most critical path in a digital circuit 750 is evaluated first.</p>

Synopsis Draft Patent Application	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 19</u></p> <p>After a cell <i>has been adjusted</i>, it becomes "<i>locked</i>" and its <i>delay cannot be</i> changed by the <i>stretching</i> algorithm.</p>	<p><u>'446 PATENT AT 15:21-25</u></p> <p>After the gate 754 <i>has been adjusted</i> to meet the Path 2 timing constraints, <i>it becomes "locked,"</i> whereby the gate 754 delay will <i>not be</i> adjusted further for the remainder of the compression and <i>stretching</i> step.</p>

1 Chris Scott Graham (State Bar No. 114498)
2 Michael N. Edelman (State Bar No. 180948)
3 **DECHERT LLP**
4 975 Page Mill Road
5 Palo Alto, California 94304
6 Telephone: (650) 813-4800
7 Facsimile: (650) 813-4848

8 Attorneys for Plaintiff SYNOPSISYS

ORIGINAL
FILED

SEP 17 2004

RICHARD W. WIEKING
CLERK, U.S. DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
SAN JOSE

9 UNITED STATES DISTRICT COURT
10 NORTHERN DISTRICT OF CALIFORNIA
11 SAN JOSE DIVISION

12 SYNOPSISYS, INC., a Delaware corporation

13 Plaintiff,

14 vs.

15 MAGMA DESIGN AUTOMATION, a
16 Delaware corporation,

17 Defendant.

CASE NO. **C04 03923 MEJ**
COMPLAINT FOR PATENT
INFRINGEMENT
ADR
DEMAND FOR JURY TRIAL
E-filing

18
19 Plaintiff SYNOPSISYS, INC. ("SYNOPSISYS") hereby alleges against Defendant MAGMA
20 DESIGN AUTOMATION ("MAGMA" or "the Defendant") as follows:

21 **JURISDICTION**

22 1. This is an action for patent infringement arising under the patent laws of the United
23 States. This Court has jurisdiction over this action under 28 U.S.C. § 1338(a).

24 **PARTIES**

25 2. SYNOPSISYS is a corporation duly organized and existing under the laws of the State
26 of Delaware, with its principal place of business in Mountain View, California.

27 ///

28 ///

3. SYNOPSISYS is informed and believes, and thereon alleges, that MAGMA is a corporation duly organized and existing under the laws of the State of Delaware, with its principal place of business in Santa Clara, California.

VENUE

4. Venue is proper in the Northern District pursuant to 28 U.S.C. § 1391(b) & (c) and 28 U.S.C. § 1400(b).

INTRADISTRICT ASSIGNMENT

5. This is an Intellectual Property Action under this Court's Assignment Plan, and therefore assignment to any division of the Court is proper pursuant to Civil L.R. 3-2(c). SYNOPSISYS believes that assignment to the San Jose division is particularly appropriate given its close proximity to the principal places of business of SYNOPSISYS and MAGMA.

FACTUAL BACKGROUND

6. On or about May 17, 1995, Lukas van Ginneken signed a Proprietary Information and Inventions Agreement (the "Agreement") as a condition to his employment by SYNOPSISYS. Paragraph 3 of this Agreement provides that all rights to any inventions made, conceived, reduced to practice or developed by van Ginneken while employed by SYNOPSISYS are automatically assigned to SYNOPSISYS. A true and correct copy of the Agreement is attached hereto as Exhibit A.

7. While employed by SYNOPSYS, van Ginneken made, conceived and developed inventions pertaining to timing closure methodology, the use of constant delay models in logic synthesis and other aspects of placement and/or synthesis. These inventions were made, conceived and developed by van Ginneken during his employment for SYNOPSYS for the purpose of developing SYNOPSYS' products, and therefore each of these inventions are encompassed by the terms of the Agreement. By operation of law, all right, title and interest to these inventions are automatically assigned to SYNOPSYS under the Agreement.

8. After leaving the employment of SYNOPSIS, van Ginneken co-founded MAGMA. Thereafter, MAGMA submitted patent applications to the Patent and Trademark Office that disclosed inventions that van Ginneken had made, conceived and developed while at SYNOPSIS, and which are owned by SYNOPSIS.

1 9. On April 23, 2002, United States Patent No. 6,378,114 ("the '114 Patent"), entitled
2 "Method for the Physical Placement of an Integrated Circuit Adaptive to Netlist Changes," was
3 issued to SYNOPSYS. van Ginneken is a named inventor on the '114 Patent. A true and correct
4 copy of the '114 Patent is attached to this complaint as Exhibit B and is incorporated by reference
5 herein.

6 10. On September 17, 2002, United States Patent No. 6,453,446 ("the '446 Patent"),
7 entitled "Timing Closure Methodology," was issued to MAGMA. The '446 Patent discloses
8 inventions which were made, conceived and developed by van Ginneken at SYNOPSYS. Pursuant
9 to the terms of the Agreement, SYNOPSYS holds legal and equitable title to the inventions in the
10 '446 Patent. A true and correct copy of the '446 Patent is attached to this complaint as Exhibit C
11 and is incorporated by reference herein.

12 11. On April 20, 2004, United States Patent No. 6,725,438 ("the '438 Patent"), entitled
13 "Timing Closure Methodology," was issued to MAGMA. The '438 Patent contains inventions
14 which were made, conceived and developed by van Ginneken at SYNOPSYS. Pursuant to the terms
15 of the Agreement, SYNOPSYS holds legal and equitable title to the inventions in the '438 Patent. A
16 true and correct copy of the '438 Patent is attached to this complaint as Exhibit D and is incorporated
17 by reference herein.

18 12. Since the issuance of the '114 Patent, '446 Patent, and '438 Patent (collectively
19 referred to hereinafter as the "SYNOPSYS PATENTS"), MAGMA has engaged in a wide range of
20 activities to infringe those patents. MAGMA has been involved in making, using, selling,
21 distributing, advertising, marketing and creating source code for products that infringe the
22 SYNOPSYS PATENTS.

23 **FIRST CAUSE OF ACTION**
24 **(PATENT INFRINGEMENT)**

25 13. SYNOPSYS is the owner of the SYNOPSYS PATENTS because, among other
26 reasons, the inventions disclosed in the patents were previously assigned to SYNOPSYS by van
27 Ginneken pursuant to the terms of the Agreement.

28 ///

1 14. MAGMA has been and still is infringing the SYNOPSIS PATENTS in violation of
2 the federal patent laws by making, using, selling, distributing, advertising, marketing and creating
3 source code for products which infringe the SYNOPSIS PATENTS. MAGMA will continue to so
4 infringe unless enjoined by this Court.

5 15. MAGMA has actively induced infringement of, or contributed to the infringement of,
6 the SYNOPSIS PATENTS under the federal patent laws by, among other things, making infringing
7 products and creating source code for infringing products and then selling, distributing, advertising
8 and marketing those infringing products to others, and will continue to do so unless enjoined by this
9 Court.

10 16. MAGMA's infringement of the SYNOPSIS PATENTS in violation of the federal
11 patent laws has been willful and deliberate, and has caused injury to SYNOPSIS.

12 17. MAGMA's infringement in violation of the federal patent laws will continue to injure
13 SYNOPSIS unless enjoined by this Court.

14 WHEREFORE, SYNOPSIS prays for judgment against the Defendant, and requests that this
15 Court impose the following remedies under the federal patent laws:

16 A. Preliminarily and permanently enjoin the Defendant from continued infringement of
17 the SYNOPSIS PATENTS, pursuant to 35 U.S.C. § 283;

18 B. Order the Defendant to account to SYNOPSIS for damages sustained by
19 SYNOPSIS as a result of the Defendant's infringement of the SYNOPSIS PATENTS, with
20 interest, pursuant to 35 U.S.C. § 284;

21 C. Order the Defendant to pay SYNOPSIS a reasonable royalty to compensate for the
22 Defendant's infringement, pursuant to 35 U.S.C. § 284;

23 D. Treble the damages resulting from the Defendant's willful and deliberate
24 infringement, pursuant to 35 U.S.C. § 284;

25 E. Award SYNOPSIS its costs, expenses and reasonable attorneys' fees incurred in
26 bringing and prosecuting this action, pursuant to 35 U.S.C. § 285;

27 ///

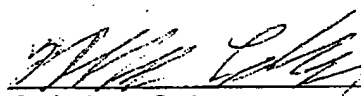
28 ///

1 F. Impose a constructive trust for the benefit of SYNOPSIS over any profits, revenues,
2 or other benefits obtained by the Defendant as a result of its infringement of the SYNOPSIS
3 PATENTS; and

4 G. Award SYNOPSIS such further relief that the Court may deem just and proper
5 arising from the Defendant's infringement of the SYNOPSIS PATENTS under the federal patent
6 laws.

7 Dated: September 17, 2004

DECHERT LLP

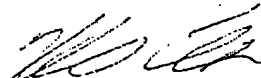
8 
9 _____
Chris Scott Graham
Michael Edelman
Attorneys for Plaintiff SYNOPSIS

DEMAND FOR JURY TRIAL

SYNOPSIS hereby demands trial by jury of all issues.

Dated: September 17, 2004

DECHERT LLP



Chris Scott Graham

Michael Edelman

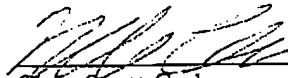
Attorneys for Plaintiff SYNOPSIS

CERTIFICATION OF INTERESTED ENTITIES OR PERSONS

Pursuant to Civil L.R. 3-16, the undersigned certifies that as of this date, other than the named parties, there is no such interest to report.

Dated: September 17, 2004

DECHERT LLP


Chris Scott Graham
Michael Edelman
Attorneys for Plaintiff SYNOPSYS

PROPRIETARY INFORMATION
AND
INVENTIONS AGREEMENT

The following confirms an agreement between me and Synopsys, Inc. (the "Company"), which is a material part of the consideration for my employment by the Company.

1. I recognize that the Company is engaged in a continuous program of research, development and production respecting its business, present and future, including fields generally related to its business and that the Company possesses and continues to possess information that has been created, discovered, developed or otherwise become known to the Company (including, without limitation, information created, discovered or developed by, or made known to, me during the period of or arising out of my employment by the Company) and/or in which property rights have been assigned, licensed or otherwise conveyed to the Company, which information has commercial value in the business in which the Company is engaged. All of the aforementioned information is hereinafter called "Proprietary Information." By way of illustration, but not limitation, Proprietary Information includes trade secrets, processes, data and know-how, computer software, improvements, inventions, works of authorship, techniques, marketing plans, strategies, forecasts and copyrightable material and customer lists.

2. I understand that my employment creates a relationship of confidence and trust between me and the Company with respect to any information:

(i) applicable to the business of the Company; or

(ii) applicable to the business of any client or customer of the Company, which may be known to me by the Company or by any client or customer of the Company, or learned by me during the period of my employment.

3. In consideration of my employment by the Company and the compensation received by me from the Company from time to time, I hereby agree as follows:

A. All Proprietary Information shall be the sole property of the Company and its assigns, and the Company and its assigns shall be sole owner of all patents, copyrights and other rights in connection therewith. I hereby assign to the Company any rights I may have or acquire in such Proprietary Information. At all times, both during my employment by the Company and after its termination, I will keep in confidence and trust all Proprietary Information, and I will not use or disclose any Proprietary Information or anything relating to it without the written consent of the Company, except as may be necessary in the ordinary course of performing my duties to the Company.

B. All documents, records, apparatus, equipment and other physical property, whether or not pertaining to Proprietary Information, furnished to me by the Company or produced by me or others in connection with my employment shall be and remain the sole property of the Company and shall be returned to the Company immediately as and when requested by the Company. Even if the Company does not so request, I shall return and deliver all such property upon termination of my employment by me or the Company for any reason and I will not take with me any such property or any reproduction of such property upon such termination.

C. I will promptly disclose to the Company, or any persons designated by it, all improvements, inventions, works of authorship, processes, techniques, know-how, formulae,

data, ideas and other information (including, without limitation, my algorithms or software), whether or not patentable, made or conceived or reduced to practice or learned by me, either alone or jointly with others, during the term of my employment (all said improvements, inventions, works of authorship, processes, techniques, know-how, formulae, data, ideas and other information shall be hereinafter collectively called "Inventions").

D. I agree that all Inventions which I make, conceive, reduce to practice or develop (in whole or in part, either alone or jointly with others) during my employment shall be the sole property of the Company to the maximum extent permitted by Section 2870 of the California Labor Code (hereinafter called "Section 2870"), a copy of which is attached hereto as Exhibit B, and to the extent permitted by law, shall be "works made for hire." The Company shall be the sole owner of all patents, copyrights, trade secret rights, rights with respect to other intellectual property or other rights in connection therewith (including, without limitation, such rights in algorithms or software). I hereby assign to the Company any rights I may have or acquire in such Inventions. I agree to perform, during and after my employment, all acts deemed necessary or desirable by the Company to permit and assist it, at the Company's expense, in obtaining, maintaining and enforcing patents, copyrights, trade secret rights, rights with respect to such Inventions and/or other Inventions I have or may at any time assign to the Company in any and all countries. Such acts may include, but are not limited to, execution of documents and assistance or cooperation in legal proceedings. I hereby irrevocably designate and appoint the Company and its duly authorized officers and agents as my agents and attorneys-in-fact to act for and on my behalf and instead of me, to execute and file any applications or related filings and to do all other lawfully permitted acts to further the prosecution, maintenance and enforcement, issuance of patents, copyrights, trade secret rights, rights with respect to mask works or other rights thereon with the same legal force and effect as if executed by me.

E. As a matter of record, I attach hereto a complete list of all inventions or improvements relevant to the subject matter of my employment by the Company which have been made by me or jointly with others prior to my employment with the Company that I desire to remove from the operation of this Agreement, and I covenant that such list is complete. If no such list is attached to this Agreement, I represent that I have no such inventions and improvements at the time of signing this Agreement.

F. During the term of my employment and for one (1) year thereafter, I will not encourage or solicit any employee of the Company to leave the Company for any reason or devote less than all of any such employee's efforts to the affairs of the Company, provided that the foregoing shall not affect any responsibility I may have as an employee if the Company with respect to the bona fide hiring and firing of Company personnel.

G. I represent that my performance of all the terms of this Agreement will not breach any agreement to keep in confidence proprietary information acquired by me in confidence or in trust prior to my employment by the Company. I have not entered into, and I agree I will not enter into, any agreement, either written or oral, in conflict herewith.

H. I represent that execution of this Agreement, my employment with the Company and my performance of my proposed duties to the Company in the development of its business will not violate any obligations I may have to my former employer.

I. This Agreement does not require assignment of an invention which an employee cannot be obligated to assign under Section 2870. However, I will disclose any Inventions as required by Section 3(c) hereof regardless of whether I believe the Invention is protected by Section 2870, in order to permit the Company to engage in a review process to determine such issues as may arise. Such disclosure shall be received in confidence by the Company.

4. This Agreement shall be effective as of the first day of my employment by the Company: June 26 1995 (date).

5. This Agreement shall be binding upon me, my heirs, executors, assigns and administrators and shall inure to the benefit of the Company, its successors and assigns.

Dated: 5/17/95


Signature

Lukas van Ginneken
(Print or Type Name)

Accepted and agreed to:

Synopsys, Inc.

By  _____

EXHIBIT A

TO

SYNOPSYS, INC. PROPRIETARY INFORMATION
AND INVENTIONS AGREEMENT

1. The following is a complete list of all inventions or improvements relevant to the subject matter of my employment by Synopsys, Inc. (the "Company") that have been made or conceived or first reduced to practice by me alone or jointly with others prior to my employment by the Company that I desire to remove from the operation of the Company's Proprietary Information and Inventions Agreement.

☐ No inventions or improvements

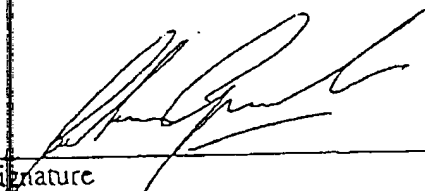
☐ See below

☒ Additional sheets attached

2. I propose to bring to my employment the following materials and documents of a former employer:

☒ No materials or documents

☐ See below


Signature

Lukas van Gluncken
Print or Type Name

EXHIBIT B

TO

SYNOPSYS, INC. PROPRIETARY INFORMATION
AND INVENTIONS AGREEMENT

Section 2870. Application of provision providing that employee shall assign or offer to assign rights in inventions to employer.

(a) Any provision in an employment agreement which provides that an employee shall assign, or offer to assign, any of his or her rights in an invention to her or her employer shall not apply to an invention that the employee entirely on his or her own time without using the employer's equipment, supplies, facilities or trade secret information except for those inventions that either:

(1) Relate at the time of conception or reduction to practice of the invention to the employer's business, or actual or demonstrably anticipated research or development by the employer;

(2) result from any work performed by the employee for the employer.

(b) To the extent a provision in an employment agreement purports to require an employee to assign an invention otherwise excluded from being required to be assigned under subdivision (a), the provision is against the public policy of this state and is unenforceable.

The following is a complete list of all inventions or improvements relevant to the subject matter of my employment by Synopsys, Inc. (the Company) that have been made or conceived or first reduced to practice by me alone or jointly with others prior to my employment by the Company that I desire to remove from the operation of the Company's Proprietary Information and Inventions Agreement.

- [0] "Efficient orthonormality testing for synthesis with pass transistor selectors" by M. R. C. M. Berkelaar and -, accepted at the International Workshop on Logic Synthesis, June 1995.
- [1] "Timing Verification and Optimization for the PowerPC Processor Family", by R.E. Mains, T. A. Mosher, - and R.F. Damiano, in: Proc. Int. Conf. on Computer Design, pp.390-393, Boston, Oct. 10-12, 1994.
- [2] "In the driver's seat of BooleDozer" by D. Brand and R.F. Damiano, -, A. D. Drumm, in: Proc. Int. Conf. on Computer Design, pp. 518-521 Boston, Oct. 10-12, 1994.
- [3] "Grammar-based optimization of synthesis scenarios" by A. Kuehlmann and -, in: Proc. Int. Conf. on Computer Design, pp. 20-25 Boston, Oct. 10-12, 1994.
- [4] "Tuning of logic synthesis scenarios" by - and A. Kuehlmann, Workshop notes of the Int. Workshop on logic synthesis, paper P7c, Tahoe City, May 23-26, 1993.
- [5] "Fanin ordering in multi-slot timing" by -, Proc. Int. Conf. on Computer Design, pp. 44-47, Cambridge, Oct. 11-14, 1992.
- [6] "The complexity of adaptive annealing" by R. H. J. M. Otten and -, Proc. Int. Conf. on Computer Design, pp. 404-407, Cambridge, Sept. 17-19, 1990.
- [7] "Buffer placement in distributed RC-tree networks for minimal Elmore delay" by -, Proc. Int. Symp. on Circuits and Systems, pp. 865-868, New Orleans, May 2-5, 1990.
- [8] "Optimal slicing of plane point placements" by - and R. H. J. M. Otten, Proc. European Design Automation Conf. pp. 322-236, Glasgow, March 12-15, 1990.
- [9] The annealing algorithm by R. H. J. M. Otten and -, ISBN 07923-9022-9, Boston:Kluwer, 1989.
- [10] The predictor-adaptor paradigm - automation of custom layout by flexible design by -, Ph.D. thesis, ISBN 90-9002703-3, Eindhoven, 1989.
- [11] "Doubly folded transistor matrix layout" by - and J. T. J. van Eijndhoven, A. H. C. M. Brouwers, Digest Int. Conf. on Computer Aided Design, Santa Clara, Nov. 7-10, 1988.
- [12] "Stop criteria in simulated annealing" R. H. J. M. Otten and -, Proc.

Int. Conf. on Computer Design, pp.549-552, Port Chester, Oct. 3-5,
1988.

[13] "An inner loop criterion for simulated annealing" by - and R.H.J.M.
Otten, Physics letters A, 130:429-435, 1988.

[14] "Soft Macro Cell generation by two dimensional folding" by - and J.
T. J. van Eijndhoven, P. R. M. van Teeffelen, T. J. Deckers, Proc. Int.
Symp. on Circuits and Systems, pp. 727-730, Espoo, June 1988.

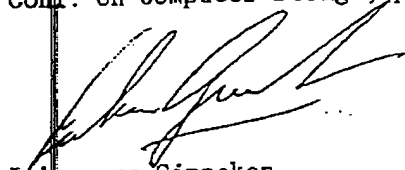
[15] "Gridless routing of general floor plans" by - and J. A. G. Jess, Digest
Int. Conf. on Computer Aided Design, pp. 30-33, Santa Clara Nov. 9-
12, 1987.

[16] "Wire planning for stackable designs", by R. K. Brayton, C. L. Chen,
J. A. G. Jess, R. H. J. M. Otten and -, Proc. Int. Symp. on VLSI tech-
nology, pp.269-273, Taipei, May 13-15, 1987.

[17] "Global wiring for custom layout design" by - and R. H. J. M. Otten,
Proc. Int. Symp. on Circuits and Systems. pp.207-208, Kyoto, June 5-
7, 1985.

[18] "Floor plan design using simulated annealing" by R. H. J. M. Otten
and -, Digest Int. Conf. on Computer Aided Design, pp. 96-98, Santa
Clara, Nov. 1984.

[19] "Stepwise layout refinement" by - and R. H. J. M. Otten, Proc. Int.
Conf. on Computer Design, pp. 30-36, Port Chester, Oct.8-11, 1984.



Lukas van Ginneken



United States Patent and Trademark Office

[Home](#) | [Site Index](#) | [Search](#) | [Guides](#) | [Contacts](#) | [eBusiness](#) | [eBiz alerts](#) | [News](#) | [Help](#)**Electronic Patent Assignment System****Confirmation Receipt**

Your assignment has been received by the USPTO.
The coversheet of the assignment is displayed below:

PATENT ASSIGNMENT

Electronic Version v1.1

Stylesheet Version v1.1

SUBMISSION TYPE:	NEW ASSIGNMENT										
NATURE OF CONVEYANCE:	ASSIGNMENT										
CONVEYING PARTY DATA											
<table border="1"><tr><th>Name</th><th>Execution Date</th></tr><tr><td>Lukas PPP van Ginneken</td><td>05/17/1995</td></tr></table>	Name	Execution Date	Lukas PPP van Ginneken	05/17/1995							
Name	Execution Date										
Lukas PPP van Ginneken	05/17/1995										
RECEIVING PARTY DATA											
<table border="1"><tr><td>Name:</td><td>Synopsys, Inc</td></tr><tr><td>Street Address:</td><td>700 E. Middlefield Road</td></tr><tr><td>City:</td><td>Mountain View</td></tr><tr><td>State/Country:</td><td>CALIFORNIA</td></tr><tr><td>Postal Code:</td><td>94043</td></tr></table>	Name:	Synopsys, Inc	Street Address:	700 E. Middlefield Road	City:	Mountain View	State/Country:	CALIFORNIA	Postal Code:	94043	
Name:	Synopsys, Inc										
Street Address:	700 E. Middlefield Road										
City:	Mountain View										
State/Country:	CALIFORNIA										
Postal Code:	94043										
PROPERTY NUMBERS Total: 3											
<table border="1"><tr><th>Property Type</th><th>Number</th></tr><tr><td>Application Number:</td><td>10828547</td></tr><tr><td>Patent Number:</td><td>6453446</td></tr><tr><td>Patent Number:</td><td>6725438</td></tr></table>	Property Type	Number	Application Number:	10828547	Patent Number:	6453446	Patent Number:	6725438			
Property Type	Number										
Application Number:	10828547										
Patent Number:	6453446										
Patent Number:	6725438										
CORRESPONDENCE DATA											
Fax Number:	(650)712-0363										
<i>Correspondence will be sent via US Mail when the fax attempt is unsuccessful.</i>											
Phone:	(650) 712-0340										
Email:	kmarley@hmbay.com										

Correspondent Name: Kenta Suzue
Address Line 1: P.O. Box 366
Address Line 2: HAYNES BEFFEL & WOLFELD
Address Line 4: Half Moon Bay, CALIFORNIA 94019

NAME OF SUBMITTER: Kenta Suzue

Signature: /ks/

Date: 01/28/2005

Total Attachments: 143

source=Complaint for Patent Infringement#page1.tif
source=Complaint for Patent Infringement#page2.tif
source=Complaint for Patent Infringement#page3.tif
source=Complaint for Patent Infringement#page4.tif
source=Complaint for Patent Infringement#page5.tif
source=Complaint for Patent Infringement#page6.tif
source=Complaint for Patent Infringement#page7.tif
source=Complaint for Patent Infringement#page8.tif
source=Complaint for Patent Infringement#page9.tif
source=Complaint for Patent Infringement#page10.tif
source=Complaint for Patent Infringement#page11.tif
source=Complaint for Patent Infringement#page12.tif
source=Complaint for Patent Infringement#page13.tif
source=Complaint for Patent Infringement#page14.tif
source=Complaint for Patent Infringement#page15.tif
source=Complaint for Patent Infringement#page16.tif
source=Complaint for Patent Infringement#page17.tif
source=Complaint for Patent Infringement#page18.tif
source=Complaint for Patent Infringement#page19.tif
source=Complaint for Patent Infringement#page20.tif
source=Complaint for Patent Infringement#page21.tif
source=Complaint for Patent Infringement#page22.tif
source=Complaint for Patent Infringement#page23.tif
source=Complaint for Patent Infringement#page24.tif
source=Complaint for Patent Infringement#page25.tif
source=Complaint for Patent Infringement#page26.tif
source=Complaint for Patent Infringement#page27.tif
source=Complaint for Patent Infringement#page28.tif
source=Complaint for Patent Infringement#page29.tif
source=Complaint for Patent Infringement#page30.tif
source=Complaint for Patent Infringement#page31.tif
source=Complaint for Patent Infringement#page32.tif
source=Complaint for Patent Infringement#page33.tif
source=Complaint for Patent Infringement#page34.tif
source=Complaint for Patent Infringement#page35.tif
source=Complaint for Patent Infringement#page36.tif
source=Complaint for Patent Infringement#page37.tif
source=Complaint for Patent Infringement#page38.tif
source=Complaint for Patent Infringement#page39.tif
source=Complaint for Patent Infringement#page40.tif
source=Complaint for Patent Infringement#page41.tif
source=Complaint for Patent Infringement#page42.tif
source=Complaint for Patent Infringement#page43.tif
source=Complaint for Patent Infringement#page44.tif

source=Complaint for Patent Infringement#page45.tif
source=Complaint for Patent Infringement#page46.tif
source=Complaint for Patent Infringement#page47.tif
source=Complaint for Patent Infringement#page48.tif
source=Complaint for Patent Infringement#page49.tif
source=Complaint for Patent Infringement#page50.tif
source=Complaint for Patent Infringement#page51.tif
source=Complaint for Patent Infringement#page52.tif
source=Complaint for Patent Infringement#page53.tif
source=Complaint for Patent Infringement#page54.tif
source=Complaint for Patent Infringement#page55.tif
source=Complaint for Patent Infringement#page56.tif
source=Complaint for Patent Infringement#page57.tif
source=Complaint for Patent Infringement#page58.tif
source=Complaint for Patent Infringement#page59.tif
source=Complaint for Patent Infringement#page60.tif
source=Complaint for Patent Infringement#page61.tif
source=Complaint for Patent Infringement#page62.tif
source=Complaint for Patent Infringement#page63.tif
source=Complaint for Patent Infringement#page64.tif
source=Complaint for Patent Infringement#page65.tif
source=Complaint for Patent Infringement#page66.tif
source=Complaint for Patent Infringement#page67.tif
source=Complaint for Patent Infringement#page68.tif
source=Complaint for Patent Infringement#page69.tif
source=Complaint for Patent Infringement#page70.tif
source=Complaint for Patent Infringement#page71.tif
source=Complaint for Patent Infringement#page72.tif
source=Complaint for Patent Infringement#page73.tif
source=Complaint for Patent Infringement#page74.tif
source=Complaint for Patent Infringement#page75.tif
source=Complaint for Patent Infringement#page76.tif
source=Complaint for Patent Infringement#page77.tif
source=Complaint for Patent Infringement#page78.tif
source=Complaint for Patent Infringement#page79.tif
source=Draft Application with Chart#page1.tif
source=Draft Application with Chart#page2.tif
source=Draft Application with Chart#page3.tif
source=Draft Application with Chart#page4.tif
source=Draft Application with Chart#page5.tif
source=Draft Application with Chart#page6.tif
source=Draft Application with Chart#page7.tif
source=Draft Application with Chart#page8.tif
source=Draft Application with Chart#page9.tif
source=Draft Application with Chart#page10.tif
source=Draft Application with Chart#page11.tif
source=Draft Application with Chart#page12.tif
source=Draft Application with Chart#page13.tif
source=Draft Application with Chart#page14.tif
source=Draft Application with Chart#page15.tif
source=Draft Application with Chart#page16.tif
source=Draft Application with Chart#page17.tif
source=Draft Application with Chart#page18.tif
source=Draft Application with Chart#page19.tif
source=Draft Application with Chart#page20.tif
source=Draft Application with Chart#page21.tif
source=Draft Application with Chart#page22.tif
source=Draft Application with Chart#page23.tif

source=Draft Application with Chart#page24.tif
source=Draft Application with Chart#page25.tif
source=Draft Application with Chart#page26.tif
source=Inv Agreement#page1.tif
source=Inv Agreement#page2.tif
source=Inv Agreement#page3.tif
source=Inv Agreement#page4.tif
source=Inv Agreement#page5.tif
source=Inv Agreement#page6.tif
source=Inv Agreement#page7.tif
source=Chart#page1.tif
source=Chart#page2.tif
source=Chart#page3.tif
source=Chart#page4.tif
source=Chart#page5.tif
source=Chart#page6.tif
source=Chart#page7.tif
source=Chart#page8.tif
source=Chart#page9.tif
source=Chart#page10.tif
source=Chart#page11.tif
source=Chart#page12.tif
source=Chart#page13.tif
source=Damiano Declaration#page1.tif
source=Damiano Declaration#page2.tif
source=Damiano Declaration#page3.tif
source=Damiano Declaration#page4.tif
source=Damiano Declaration#page5.tif
source=Damiano Declaration#page6.tif
source=Damiano Declaration#page7.tif
source=Damiano Declaration#page8.tif
source=Damiano Declaration#page9.tif
source=Damiano Declaration#page10.tif
source=Damiano Declaration#page11.tif
source=Damiano Declaration#page12.tif
source=Damiano Declaration#page13.tif
source=Damiano Declaration#page14.tif
source=Damiano Declaration#page15.tif
source=Damiano Declaration#page16.tif
source=Damiano Declaration#page17.tif
source=Damiano Declaration#page18.tif

RECEIPT INFORMATION

EPAS ID:	PAT20728
Receipt Date:	01/28/2005
Fee Amount:	\$120

[Return to home page](#)

[| .HOME](#) | [INDEX](#) | [SEARCH](#) | [eBUSINESS](#) | [CONTACT US](#) | [PRIVACY STATEMENT](#)

1 Chris Scott Graham (State Bar No. 114498)
2 Michael N. Edelman (State Bar No. 180948)
3 **DECHERT LLP**
4 975 Page Mill Road
5 Palo Alto, California 94304
6 Telephone: (650) 813-4800
7 Facsimile: (650) 813-4848

8 Attorneys for Plaintiff SYNOPSISYS

**ORIGINAL
FILED**

SEP 17 2004

RICHARD W. WIEKING
CLERK, U.S. DISTRICT COURT
NORTHERN DISTRICT OF CALIFORNIA
SAN JOSE

9 UNITED STATES DISTRICT COURT
10 NORTHERN DISTRICT OF CALIFORNIA
11 SAN JOSE DIVISION

12 SYNOPSISYS, INC., a Delaware corporation

13 Plaintiff,

14 vs.

15 MAGMA DESIGN AUTOMATION, a
16 Delaware corporation,

17 Defendant.

C04 03923 MEJ
ADR
E-filing
CASE NO.
COMPLAINT FOR PATENT
INFRINGEMENT
DEMAND FOR JURY TRIAL

18
19 Plaintiff SYNOPSISYS, INC. ("SYNOPSISYS") hereby alleges against Defendant MAGMA
20 DESIGN AUTOMATION ("MAGMA" or "the Defendant") as follows:

21 **JURISDICTION**

22 1. This is an action for patent infringement arising under the patent laws of the United
23 States. This Court has jurisdiction over this action under 28 U.S.C. § 1338(a).

24 **PARTIES**

25 2. SYNOPSISYS is a corporation duly organized and existing under the laws of the State
26 of Delaware, with its principal place of business in Mountain View, California.

27 ///

28 ///

3. SYNOPSISYS is informed and believes, and thereon alleges, that MAGMA is a corporation duly organized and existing under the laws of the State of Delaware, with its principal place of business in Santa Clara, California.

VENUE

4. Venue is proper in the Northern District pursuant to 28 U.S.C. § 1391(b) & (c) and 28 U.S.C. § 1400(b).

INTRADISTRICT ASSIGNMENT

5. This is an Intellectual Property Action under this Court's Assignment Plan, and therefore assignment to any division of the Court is proper pursuant to Civil L.R. 3-2(c). SYNOPSISYS believes that assignment to the San Jose division is particularly appropriate given its close proximity to the principal places of business of SYNOPSISYS and MAGMA.

FACTUAL BACKGROUND

6. On or about May 17, 1995, Lukas van Ginneken signed a Proprietary Information and Inventions Agreement (the "Agreement") as a condition to his employment by SYNOPSISYS. Paragraph 3 of this Agreement provides that all rights to any inventions made, conceived, reduced to practice or developed by van Ginneken while employed by SYNOPSISYS are automatically assigned to SYNOPSISYS. A true and correct copy of the Agreement is attached hereto as Exhibit A.

7. While employed by SYNOPSIS, van Ginneken made, conceived and developed inventions pertaining to timing closure methodology, the use of constant delay models in logic synthesis and other aspects of placement and/or synthesis. These inventions were made, conceived and developed by van Ginneken during his employment for SYNOPSIS for the purpose of developing SYNOPSIS' products, and therefore each of these inventions are encompassed by the terms of the Agreement. By operation of law, all right, title and interest to these inventions are automatically assigned to SYNOPSIS under the Agreement.

8. After leaving the employment of SYNOPSYS, van Ginneken co-founded MAGMA. Thereafter, MAGMA submitted patent applications to the Patent and Trademark Office that disclosed inventions that van Ginneken had made, conceived and developed while at SYNOPSYS, and which are owned by SYNOPSYS.

1 9. On April 23, 2002, United States Patent No. 6,378,114 ("the '114 Patent"), entitled
2 "Method for the Physical Placement of an Integrated Circuit Adaptive to Netlist Changes," was
3 issued to SYNOPSYS. van Ginneken is a named inventor on the '114 Patent. A true and correct
4 copy of the '114 Patent is attached to this complaint as Exhibit B and is incorporated by reference
5 herein.

6 10. On September 17, 2002, United States Patent No. 6,453,446 ("the '446 Patent"),
7 entitled "Timing Closure Methodology," was issued to MAGMA. The '446 Patent discloses
8 inventions which were made, conceived and developed by van Ginneken at SYNOPSYS. Pursuant
9 to the terms of the Agreement, SYNOPSYS holds legal and equitable title to the inventions in the
10 '446 Patent. A true and correct copy of the '446 Patent is attached to this complaint as Exhibit C
11 and is incorporated by reference herein.

12 11. On April 20, 2004, United States Patent No. 6,725,438 ("the '438 Patent"), entitled
13 "Timing Closure Methodology," was issued to MAGMA. The '438 Patent contains inventions
14 which were made, conceived and developed by van Ginneken at SYNOPSYS. Pursuant to the terms
15 of the Agreement, SYNOPSYS holds legal and equitable title to the inventions in the '438 Patent. A
16 true and correct copy of the '438 Patent is attached to this complaint as Exhibit D and is incorporated
17 by reference herein.

18 12. Since the issuance of the '114 Patent, '446 Patent, and '438 Patent (collectively
19 referred to hereinafter as the "SYNOPSYS PATENTS"), MAGMA has engaged in a wide range of
20 activities to infringe those patents. MAGMA has been involved in making, using, selling,
21 distributing, advertising, marketing and creating source code for products that infringe the
22 SYNOPSYS PATENTS.

23 **FIRST CAUSE OF ACTION**
24 **(PATENT INFRINGEMENT)**

25 13. SYNOPSYS is the owner of the SYNOPSYS PATENTS because, among other
26 reasons, the inventions disclosed in the patents were previously assigned to SYNOPSYS by van
27 Ginneken pursuant to the terms of the Agreement.

28 ///

1 14. MAGMA has been and still is infringing the SYNOPSIS PATENTS in violation of
2 the federal patent laws by making, using, selling, distributing, advertising, marketing and creating
3 source code for products which infringe the SYNOPSIS PATENTS. MAGMA will continue to so
4 infringe unless enjoined by this Court.

5 15. MAGMA has actively induced infringement of, or contributed to the infringement of,
6 the SYNOPSIS PATENTS under the federal patent laws by, among other things, making infringing
7 products and creating source code for infringing products and then selling, distributing, advertising
8 and marketing those infringing products to others, and will continue to do so unless enjoined by this
9 Court.

10 16. MAGMA's infringement of the SYNOPSIS PATENTS in violation of the federal
11 patent laws has been willful and deliberate, and has caused injury to SYNOPSIS.

12 17. MAGMA's infringement in violation of the federal patent laws will continue to injure
13 SYNOPSIS unless enjoined by this Court.

14 WHEREFORE, SYNOPSIS prays for judgment against the Defendant, and requests that this
15 Court impose the following remedies under the federal patent laws:

16 A. Preliminarily and permanently enjoin the Defendant from continued infringement of
17 the SYNOPSIS PATENTS, pursuant to 35 U.S.C. § 283;

18 B. Order the Defendant to account to SYNOPSIS for damages sustained by
19 SYNOPSIS as a result of the Defendant's infringement of the SYNOPSIS PATENTS, with
20 interest, pursuant to 35 U.S.C. § 284;

21 C. Order the Defendant to pay SYNOPSIS a reasonable royalty to compensate for the
22 Defendant's infringement, pursuant to 35 U.S.C. § 284;

23 D. Treble the damages resulting from the Defendant's willful and deliberate
24 infringement, pursuant to 35 U.S.C. § 284;

25 E. Award SYNOPSIS its costs, expenses and reasonable attorneys' fees incurred in
26 bringing and prosecuting this action, pursuant to 35 U.S.C. § 285;

27 ///

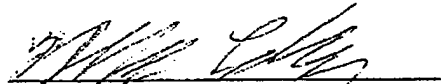
28 ///

1 F. Impose a constructive trust for the benefit of SYNOPSIS over any profits, revenues,
2 or other benefits obtained by the Defendant as a result of its infringement of the SYNOPSIS
3 PATENTS; and

4 G. Award SYNOPSIS such further relief that the Court may deem just and proper
5 arising from the Defendant's infringement of the SYNOPSIS PATENTS under the federal patent
6 laws.

7 Dated: September 17, 2004

DECHERT LLP

8 

9 Chris Scott Graham

10 Michael Edelman

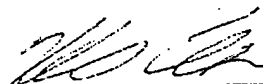
11 Attorneys for Plaintiff SYNOPSIS
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

DEMAND FOR JURY TRIAL

SYNOPSIS hereby demands trial by jury of all issues.

Dated: September 17, 2004

DECHERT LLP



Chris Scott Graham

Michael Edelman

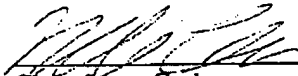
Attorneys for Plaintiff SYNOPSIS

CERTIFICATION OF INTERESTED ENTITIES OR PERSONS

Pursuant to Civil L.R. 3-16, the undersigned certifies that as of this date, other than the named parties, there is no such interest to report.

Dated: September 17, 2004

DECHERT LLP



Chris Scott Graham

Michael Edelman

Attorneys for Plaintiff SYNOPSYS

PROPRIETARY INFORMATION
AND
INVENTIONS AGREEMENT

The following confirms an agreement between me and Synopsys, Inc. (the "Company"), which is a material part of the consideration for my employment by the Company.

1. I recognize that the Company is engaged in a continuous program of research, development and production respecting its business, present and future, including fields generally related to its business and that the Company possesses and continues to possess information that has been created, discovered, developed or otherwise become known to the Company (including, without limitation, information created, discovered or developed by, or made known to, me during the period of or arising out of my employment by the Company) and/or in which property rights have been assigned, licensed or otherwise conveyed to the Company, which information has commercial value in the business in which the Company is engaged. All of the aforementioned information is hereinafter called "Proprietary Information." By way of illustration, but not limitation, Proprietary Information includes trade secrets, processes, data and know-how, computer software, improvements, inventions, works of authorship, techniques, marketing plans, strategies, forecasts and copyrightable material and customer lists.

2. I understand that my employment creates a relationship of confidence and trust between me and the Company with respect to any information:

(i) applicable to the business of the Company; or

(ii) applicable to the business of any client or customer of the Company, which may be known to me by the Company or by any client or customer of the Company, or learned by me during the period of my employment.

3. In consideration of my employment by the Company and the compensation received by me from the Company from time to time, I hereby agree as follows:

A. All Proprietary Information shall be the sole property of the Company and its assigns, and the Company and its assigns shall be sole owner of all patents, copyrights and other rights in connection therewith. I hereby assign to the Company any rights I may have or acquire in such Proprietary Information. At all times, both during my employment by the Company and after its termination, I will keep in confidence and trust all Proprietary Information, and I will not use or disclose any Proprietary Information or anything relating to it without the written consent of the Company, except as may be necessary in the ordinary course of performing my duties to the Company.

B. All documents, records, apparatus, equipment and other physical property, whether or not pertaining to Proprietary Information, furnished to me by the Company or produced by me or others in connection with my employment shall be and remain the sole property of the Company and shall be returned to the Company immediately as and when requested by the Company. Even if the Company does not so request, I shall return and deliver all such property upon termination of my employment by me or the Company for any reason and I will not take with me any such property or any reproduction of such property upon such termination.

C. I will promptly disclose to the Company, or any persons designated by it, all improvements, inventions, works of authorship, processes, techniques, know-how, formulae,

data, ideas and other information (including, without limitation, my algorithms or software), whether or not patentable, made or conceived or reduced to practice or learned by me, either alone or jointly with others, during the term of my employment (all said improvements, inventions, works of authorship, processes, techniques, know-how, formulae, data, ideas and other information shall be hereinafter collectively called "Inventions").

D. I agree that all Inventions which I make, conceive, reduce to practice or develop (in whole or in part, either alone or jointly with others) during my employment shall be the sole property of the Company to the maximum extent permitted by Section 2870 of the California Labor Code (hereinafter called "Section 2870"), a copy of which is attached hereto as Exhibit B, and to the extent permitted by law, shall be "works made for hire." The Company shall be the sole owner of all patents, copyrights, trade secret rights, rights with respect to other intellectual property or other rights in connection therewith (including, without limitation, such rights in algorithms or software). I hereby assign to the Company any rights I may have or acquire in such Inventions. I agree to perform, during and after my employment, all acts deemed necessary or desirable by the Company to permit and assist it, at the Company's expense, in obtaining, maintaining and enforcing patents, copyrights, trade secret rights, rights with respect to such Inventions and/or other Inventions I have or may at any time assign to the Company in any and all countries. Such acts may include, but are not limited to, execution of documents and assistance or cooperation in legal proceedings. I hereby irrevocably designate and appoint the Company and its duly authorized officers and agents as my agents and attorneys-in-fact to act for and on my behalf and instead of me, to execute and file any applications or related filings and to do all other lawfully permitted acts to further the prosecution, maintenance and enforcement, issuance of patents, copyrights, trade secret rights, rights with respect to mask works or other rights thereon with the same legal force and effect as if executed by me.

E. As a matter of record, I attach hereto a complete list of all inventions or improvements relevant to the subject matter of my employment by the Company which have been made by me or jointly with others prior to my employment with the Company that I desire to remove from the operation of this Agreement, and I covenant that such list is complete. If no such list is attached to this Agreement, I represent that I have no such inventions and improvements at the time of signing this Agreement.

F. During the term of my employment and for one (1) year thereafter, I will not encourage or solicit any employee of the Company to leave the Company for any reason or devote less than all of any such employee's efforts to the affairs of the Company, provided that the foregoing shall not affect any responsibility I may have as an employee if the Company with respect to the bona fide hiring and firing of Company personnel.

G. I represent that my performance of all the terms of this Agreement will not breach any agreement to keep in confidence proprietary information acquired by me in confidence or in trust prior to my employment by the Company. I have not entered into, and I agree I will not enter into, any agreement, either written or oral, in conflict herewith.

H. I represent that execution of this Agreement, my employment with the Company and my performance of my proposed duties to the Company in the development of its business will not violate any obligations I may have to my former employer.

I. This Agreement does not require assignment of an invention which an employee cannot be obligated to assign under Section 2870. However, I will disclose any Inventions as required by Section 3(c) hereof regardless of whether I believe the Invention is protected by Section 2870, in order to permit the Company to engage in a review process to determine such issues as may arise. Such disclosure shall be received in confidence by the Company.

4. This Agreement shall be effective as of the first day of my employment by the Company: June 26 1995 (date).

5. This Agreement shall be binding upon me, my heirs, executors, assigns and administrators and shall inure to the benefit of the Company, its successors and assigns.

Dated: 5/17/95


Signature

Lukas van Ginneken
(Print or Type Name)

Accepted and agreed to:

Synopsys, Inc.

By  _____

EXHIBIT A

TO

SYNOPSYS, INC. PROPRIETARY INFORMATION
AND INVENTIONS AGREEMENT

1. The following is a complete list of all inventions or improvements relevant to the subject matter of my employment by Synopsys, Inc. (the "Company") that have been made or conceived or first reduced to practice by me alone or jointly with others prior to my employment by the Company that I desire to remove from the operation of the Company's Proprietary Information and Inventions Agreement.

☐ No inventions or improvements

☐ See below

☒

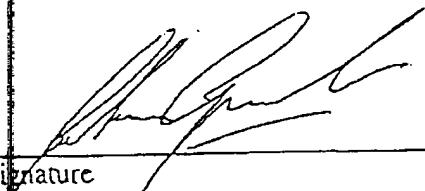
Additional sheets attached

2. I propose to bring to my employment the following materials and documents of a former employer:

☒

No materials or documents

☐ See below


Signature

Lukas van Glnneken,
Print or Type Name

EXHIBIT B

TO

SYNOPSYS, INC. PROPRIETARY INFORMATION
AND INVENTIONS AGREEMENT

Section 2870. Application of provision providing that employee shall assign or offer to assign rights in inventions to employer.

(a) Any provision in an employment agreement which provides that an employee shall assign, or offer to assign, any of his or her rights in an invention to her or her employer shall not apply to an invention that the employee entirely on his or her own time without using the employer's equipment, supplies, facilities or trade secret information except for those inventions that either:

(1) Relate at the time of conception or reduction to practice of the invention to the employer's business, or actual or demonstrably anticipated research or development by the employer;

(2) result from any work performed by the employee for the employer.

(b) To the extent a provision in an employment agreement purports to require an employee to assign an invention otherwise excluded from being required to be assigned under subdivision (a), the provision is against the public policy of this state and is unenforceable.

The following is a complete list of all inventions or improvements relevant to the subject matter of my employment by Synopsys, Inc. (the Company) that have been made or conceived or first reduced to practice by me alone or jointly with others prior to my employment by the Company that I desire to remove from the operation of the Company's Proprietary Information and Inventions Agreement.

- [0] "Efficient orthonormality testing for synthesis with pass transistor selectors" by M. R. C. M. Berkelaar and -, accepted at the International Workshop on Logic Synthesis, June 1995.
- [1] "Timing Verification and Optimization for the PowerPC Processor Family", by R.E. Mains, T. A. Mosher, - and R.F. Damiano, in: Proc. Int. Conf. on Computer Design, pp.390-393, Boston, Oct. 10-12, 1994.
- [2] "In the driver's seat of BooleDozer" by D. Brand and R.F. Damiano, -, A. D. Drumm, in: Proc. Int. Conf. on Computer Design, pp. 518-521 Boston, Oct. 10-12, 1994.
- [3] "Grammar-based optimization of synthesis scenarios" by A. Kuehlmann and -, in: Proc. Int. Conf. on Computer Design, pp. 20-25 Boston, Oct. 10-12, 1994.
- [4] "Tuning of logic synthesis scenarios" by - and A. Kuehlmann, Workshop notes of the Int. Workshop on logic synthesis, paper P7c, Tahoe City, May 23-26, 1993.
- [5] "Fanin ordering in multi-slot timing" by -, Proc. Int. Conf. on Computer Design, pp. 44-47, Cambridge, Oct. 11-14, 1992.
- [6] "The complexity of adaptive annealing" by R. H. J. M. Otten and -, Proc. Int. Conf. on Computer Design, pp. 404-407, Cambridge, Sept. 17-19, 1990.
- [7] "Buffer placement in distributed RC-tree networks for minimal Elmore delay" by -, Proc. Int. Symp. on Circuits and Systems, pp. 865-868, New Orleans, May 2-5, 1990.
- [8] "Optimal slicing of plane point placements" by - and R. H. J. M. Otten, Proc. European Design Automation Conf. pp. 322-236, Glasgow, March 12-15, 1990.
- [9] The annealing algorithm by R. H. J. M. Otten and -, ISBN 07923-9022-9, Boston:Kluwer, 1989.
- [10] The predictor-adaptor paradigm - automation of custom layout by flexible design by -, Ph.D. thesis, ISBN 90-9002703-3, Eindhoven, 1989.
- [11] "Doubly folded transistor matrix layout" by - and J. T. J. van Eijndhoven, A. H. C. M. Brouwers, Digest Int. Conf. on Computer Aided Design, Santa Clara, Nov. 7-10, 1988.
- [12] "Stop criteria in simulated annealing" R. H. J. M. Otten and -, Proc.

Int. Conf. on Computer Design, pp.549-552, Port Chester, Oct. 3-5,
1988.

[13] "An inner loop criterion for simulated annealing" by - and R.H.J.M.
Otten, Physics letters A, 130:429-435, 1988.

[14] "Soft Macro Cell generation by two dimensional folding" by - and J.
T. J. van Eijndhoven, P. R. M. van Teeffelen, T. J. Deckers, Proc. Int.
Symp. on Circuits and Systems, pp. 727-730, Espoo, June 1988.

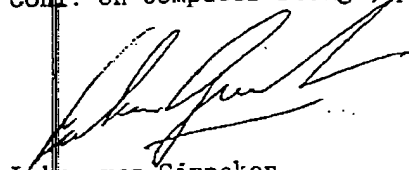
[15] "Gridless routing of general floor plans" by - and J. A. G. Jess, Digest
Int. Conf. on Computer Aided Design, pp. 30-33, Santa Clara Nov. 9-
12, 1987.

[16] "Wire planning for stackable designs", by R. K. Brayton, C. L. Chen,
J. A. G. Jess, R. H. J. M. Otten and -, Proc. Int. Symp. on VLSI tech-
nology, pp.269-273, Taipeh, May 13-15, 1987.

[17] "Global wiring for custom layout design" by - and R. H. J. M. Otten,
Proc. Int. Symp. on Circuits and Systems. pp.207-208, Kyoto, June 5-
7, 1985.

[18] "Floor plan design using simulated annealing" by R. H. J. M. Otten
and -, Digest Int. Conf. on Computer Aided Design, pp. 96-98, Santa
Clara, Nov. 1984.

[19] "Stepwise layout refinement" by - and R. H. J. M. Otten, Proc. Int.
Conf. on Computer Design, pp. 30-36, Port Chester, Oct.8-11, 1984.



Lukas van Ginneken

SYNOPSYS CONFIDENTIAL

APPLICATION FOR UNITED STATES PATENT

in the name of

LUKAS PAUL PIETER PEPIJN VAN GINNEKEN

of

SYNOPSYS, INC.

for

METHOD FOR ACHIEVING TIMING CLOSURE OF DIGITAL NETWORKS

AND

METHOD FOR AREA OPTIMIZATION OF DIGITAL NETWORKS UNDER TIMING CLOSURE

class 364/489

BACKGROUND OF THE INVENTION

This application relates to a method for achieving timing closure of digital networks consisting of structuring and mapping and a method for area optimization of digital networks using placement and sizing, while maintaining timing closure.

(Prior Art)

Figure 1 shows the conventional approach to digital network synthesis.

Digital network synthesis is a process in which computer programs optimizes digital networks. At the beginning of the synthesis process, a human designer 110 specifies a design 111 at a high level of abstraction using a high level design specification language, such as Verilog or VHDL.

In step 101 the abstract network specification of the design is transformed into an unmapped digital network representation in

LMAORN.P50

memory
by a computer program,
such as HDL Compiler, available from Synopsys Inc. of Mountain
View,
California.

In step 102 Logic synthesis algorithms
optimize the network by changing the structure of the network
without
changing the function of the network.

In step 103, logic
network synthesis algorithms map the abstract network
representation to cells from the library.

Step 104
optimizes the size of the cells. The size of the cell, together
with
the load of the cell determines the delay of the cell. The sizing
algorithm adjusts the sizes of the cells. Changing the sizes of
the cells
affects delay and area, and the sizing algorithm manipulates the
cell sizes so as to
minimize delay and area.

Steps 102, 103 and 104 are performed by a computer
program, such as "Design Compiler" TM available from Synopsys
Inc., of
Mountain View, California.

Step 105 determines the placement of the cells on the chip.
Placement algorithms
attempt to keep the length of the nets short, as longer nets need
more
area on the chip and the increased net load of longer nets will
make the
network slower. The network remains unchanged during the
placement.

Finally step 106 determines the exact routing of the nets on the
chip.
Steps 105 and 106 are done by a computer program, such as "Cell
Ensemble" TM
available from Cadence Inc. of San Jose, California.

(Problems with prior art)

The major problem with the conventional approach is that the net
length and hence the cell delay is not known until after
placement. Before placement, net length must be estimated. This
is
usually done with an estimation function or table which gives the

load
of a net based on its fanout. Experience has shown that it is
very
difficult to estimate the length of the nets accurately.
Essentially
net length behaves as a random variable.

The result is unpleasant surprises after placement step 105. Some
nets turn out
to be longer then expected, and because of the longer delays the
timing
constraints are not met. Timing closure is not certain until
after step 105.

If timing closure is not achieved the options the designer has
are
expensive and unreliable. He may choose to fix the design
manually,
which is difficult and time consuming, because the automatically
optimized network is hard to understand. He may choose to change
his
HDL specification and repeat the synthesis process. Again timing
closure
will not be certain until after placement, which means that the
entire
process needs to be traversed before the designer knows if his
HDL
changes were successful.

A common method of dealing with inaccurate net load estimates is
to use
net load estimates which are considerably larger then accurate
estimates. This causes the sizes of the cells to be considerably
larger then necessary but reduces the probability of not meeting
the
timing constraints after placement. Clearly using cells with
sizes which
are larger then necessary is wasteful in both silicon area and
power
consumption. The chips thus synthesized will be larger, cost more
to
produce and use more electrical power then necessary.

A second problem with the conventional approach is that the
effect
of synthesis decisions is hard to calculate. Performing timing
analysis
during optimization is very time consuming, and accounts for most
of the run time of conventional synthesis systems.

In step 103 it is difficult to take decisions based on delay
without

knowing load and size of the cells as well.

In step 104, changing the size of a cell affects the loads of the fanin cells, and thus the delay of the fanin cells. In more complex delay models, which take into account the transition time of the signals, also the delay of the fanout cells is affected. Usually the size parameter cannot have any arbitrary value: Because the library of cells has been designed before the network synthesis started, only a few sizes are available. 3 or 4 sizes per cell is common. This makes it harder to find a good solution.

In step 105 the placement program will modify the net lengths. Depending on which location was chosen for each cell, the length of each net can be different. As the length differs, the capacitive load of the net changes, and as a result, the delay of the cell driving the net changes. Therefore the delays which were carefully optimized during the logic synthesis, are very different after placement, and the optimization of the network is not very good.

Much of the progress in the state of the art can be characterized as increased integration. This is represented in figure 1 as various feedback paths, which repeat and alternate steps. The general direction has been towards programs which do structuring, mapping, sizing and placement simultaneously. It has lead to increasingly complex software systems which are slow and difficult to design and maintain.

Iterating between placement and sizing has been especially hard to execute because placement programs are not sold by the same design automation software vendors as logic synthesis programs. Also they are not run by the same users: the logic synthesis program is often run by the designer, who also wrote the HDL specification. The placement program is often run by the silicon chip manufacturer, after the design is considered complete.

SUMMARY OF THE INVENTION

(objectives)

It is an object of the present invention to achieve timing closure as quickly as possible in the synthesis process. This will give the human designer early warning if the design is infeasible, because it is over constrained.

The invention achieves this object by

- speeding up the algorithms in the synthesis process
- performing the feasibility check early in the synthesis process, that is, before placement, instead of after placement.
- maintaining feasibility throughout the remainder of the synthesis process, so that it can be guaranteed to succeed and can be executed automatically.

The present invention overcomes the problems of the conventional approach

by not choosing a size for a cell at all. Rather than choosing a default size, as conventional methods do, we choose a delay and let the size implicitly be whatever it needs to be to meet that delay.

In the conventional method of optimization, the structure, mapping, size and placement are chosen to optimize delay and area. In our formulation of the problem, we choose the structure, mapping, delay and placement, to optimize size and area. In our formulation, size only affects the area, so area only remains as an optimization goal.

The present invention speeds up the programs by simplifying or eliminating timing analysis. While optimizing network delay in order to achieve timing closure, the delays are constant, which will speed up delay calculation. Also, since changes to the network do not change the delay of the cells, the amount of recalculation is drastically reduced. While optimizing area after timing closure has been achieved, timing analysis is not needed, as the delay of the cells does not change. The one step

where timing analysis using complex delay models is necessary is in the stretching step. Here too, library design rules are taken into account.

The present invention maintains timing closure after it has been achieved by adjusting the size of the cell during or after placement. The adjustments compensate for the fact that the placement algorithm can assign different net lengths to different nets and that these lengths are difficult to predict before placement.

(language of the main claims)

In accordance with the purpose of this invention, as embodied and broadly described herein, the invention is a method for the structuring and mapping of an unmapped digital network comprising the following steps:

- a) structuring of the digital network, using network slack as an optimization goal, where network slack is calculated assuming that the delay of the cells of the network is constant with respect to load.
- b) mapping of the cells of the network, using network slack as an optimization goal, where network slack is calculated assuming that the delay of the cells of the network is constant with respect to load.
- c) estimation of the area of the network based on net load

In further accordance with the purpose of this invention, as embodied

and broadly described herein, the invention is a method for the placement and sizing of cells of a mapped digital network, the method comprising the steps of:

- d) Choosing a target delay for each cell.
- e) Computing the network slack using the target delays.
- f) Placement of the cells of the network.
- g) Sizing of the cells of the network such that the network meets the network slack as computed by step b).

In further accordance with the purpose of this invention, as embodied

and broadly described herein, the invention is a method for the placement of the cells of a digital network, the method comprising the steps, of:

- f1) The calculation of net weights that reflect the change of network area

LMAORN.P50

due to sizing as a function of net length.
f2) Placement of the cells of the network
where the weighted network net length is used as
a placement objective, the weighted network net length being the
sum
of the weighted net lengths of all nets, each net length being
multiplied by a weight,

Objects and advantages of the invention will be set forth in part
in
the description which follows and in part will be obvious from
the
description or may be learned by practice of the invention. The
objects and advantages of the invention will be realized and
attained
by means of the elements and combinations particularly pointed
out in
the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and
constitute
part of this specification, illustrate several embodiments of the
invention and, together with the description, serve to explain
the
principles of the invention.

Fig. 1 is a flow chart showing the flow of the conventional
method.

Fig. 2 is a flow chart showing the flow according to the present
invention.

Fig. 3 is a block diagram of a computer.

Fig. 4 is a schematic diagram of a digital network.

Fig. 5 is a schematic diagram of the electronic and the physical
implementations
of a cell.

Fig. 6 is a timing diagram illustrating the concept of slack and
other timing concepts.

Fig. 7 is a graph showing the relationship between the delay of a
cell, the
size of a cell and the load of a cell.

LMAORN.P50

Fig. 10 is a

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Reference will now be made in detail to the preferred embodiments of the invention, examples of which are illustrated in the accompanying drawings. Wherever possible, the same reference numbers will be used throughout the drawings to refer to the same or like parts.

(what insight led to this invention?)

(Software patent application)

The preferred embodiment of the present invention is as one or more computer programs. The digital network, its elements and attributes, exist during the process of the method only as data structures in the memory of the computer. Methods in prior art are known to persons of ordinary skill in the art to convert the design data in memory eventually to an actual physical implementation of the network.

(Computer system)

Figure 1 is a block diagram of a computer system 100 in accordance with the present invention. Computer system 100 includes a central processing unit 101, bus 102, memory 103, input device 104 and output device 105. It will be understood by a person of ordinary skill in the art that computer system 100 can also include numerous elements not shown in the figure for the sake of clarity, such as disk drives, tape drives, mice, printers, network connections, additional CPUs, etc. Memory 103 contains a program 107, which embodies the invention, and a data structure representation of the network 106.

(Network terminology)

Figure 2 is a schematic diagram of a digital network. Digital *network* 200 is composed of a plurality of *cells* 205, 206, 207, 208,

LMAORN.P50

209,
 connected by a plurality of *nets*. Each cell (e.g. 208) has one
 or
 more inputs 212, 213, and a single output 214. Each net is
 connected
 to one output and one or more inputs. The cells can be
 combinational
 gates 207, 208, 209, whose function is represented as an
 expression
 in the Boolean algebra, using AND, OR and NOT operators, or the
 cells
 can be registers 205, 206. All feed back loops 210, 211 in the
 network contain at least one register. Cells whose outputs are
 connected to the inputs of a cell are called the fanin of the
 latter
 cell. Cells whose inputs are connected to the output of a cell
 are
 called the fanout of the latter cell.

(Computation)

The digital network performs a logic *function* by processing
 digital
 binary input data in a number of cycles. The input data is
 presented
 to the network on its *primary inputs* 201, 202, and the result
 of the
 computation of the network function is presented at the *primary
 outputs* 203, 204, of the network. The computation of the
 function
 takes one or more cycles. During each cycle the gate functions
 are
 calculated. The results are stored in the registers for use in
 the
 next cycle. The data flow is from inputs to outputs and will be
 assumed to be pictured as going from left to right throughout
 this
 text.

(Mapping)

Each cell can be *mapped* to a *book* in which case a electronic
 realization in
 terms of transistors has been chosen, and physical attributes
 such as
 delay are known. If a cell is not mapped to a book it is
 unmapped, in which
 case no electronic realization has been selected, and the
 function of
 the cell is only known in abstract terms, such as Boolean
 algebraic
 expressions. A *library* of books is designed in advance, before

synthesis starts. These books are of generic types, and can be used to build arbitrary designs.

(Size)

The cell has a delay, an area and its input pins have an input pin capacitance. The **size** of the cell is a multiplier which is applied to equally to all transistor channel widths in the electronic circuit of the cell. Thus size of a cell is a scale factor which is used to scale its output load driving capability (see below), its area, and its input pin loads.

(Timing constraints)

Each primary input or primary output has an associated delay, called the **input delay**, respectively **output delay**, which represents delays external to the network. The network needs a certain amount of time to perform one cycle, called the **cycle delay**. Together the cycle delay, the input delays and the output delays form the timing constraints of the network. Meeting the timing constraints is called **timing closure** and it is a major objective of the synthesis process.

(Timing)

The delay of a path is measured as the sum of the gate delays over said path from begin point to end point. The cycle delay is the maximum of all path delays. Primary outputs and register inputs are timing path end points. Primary inputs and register outputs are timing path begin points.

(Slack)

The **arrival time** of the data at a gate is computed by taking the maximum arrival time of its fanin cells each increased by the delay from the input pin to the output pin. The arrival times can be computed by traversing the network from left to right, that is, from timing begin points to timing end points. Similarly the **required time** of the data at a gate can be computed by traversing the network from end points to begin points. The required time is the minimum required time of its fanout cells, each reduced by the delay from

the
input to the output pin of that fanout cell.

The difference between the required time and the arrival time is the
slack. If the arrival time is smaller than the required time, the
timing constraints are met, and the slack is positive. If the arrival
time is larger than the required time, the timing constraints are not
met, and the slack is negative. The arrival time and required
time may be different depending on whether the data is zero (0) or one
(1). There also may be multiple arrival times and multiple
required times
to model a variety of timing constraints. All slacks can be
summarized
as a single worst slack number, called the *network slack*.
Timing closure
is achieved if the network slack is non-negative.

(Delay model)

The delay D of a gate depends on many factors, among them its
function, its size S and the capacitive load C of the gate. The
delay
may be different for different inputs of the gate and it may be
different for the falling and the rising transition. It is
important
to note that the dependency on size and load can be captured as
the
dependency on a single parameter C/S, and the delay D is
non-negative
and monotonically increasing with C/S.

$$\begin{aligned}C &> 0 \\S &> 0 \\D &= f(C/S) \\f(C/S) &> 0 \\f'(C/S) &> 0\end{aligned}$$

Figure 4 illustrates the relationship between the three
variables,
size S, delay D and capacitive load C. Each of the three planes
shows
the relationship between two variables, while the third variable
is
constant, (not necessarily zero).

(Detailed description)

The preferred embodiment of the invention is a software program that can be stored in the memory of a computer, and can be executed by the central processing unit of the computer so that the computer performs the method described herein.

The software program consists of many parts or subprograms which together perform the method described in this invention. The essence of the invention is that logic synthesis is done in a size independent way, and that sizes are determined after placement, and that we guarantee that the delay numbers before placement can be met by sizing after placement.

The preferred embodiment consists of three parts: (See fig...) to wit

- a) The logic synthesis program
- b) The placement program
- c) The sizing program

- analyze the library
- read logic
- library independent optimization.

The first step is to analyze the library that will be used for logic synthesis. The library contains the cells that will be used to implement the logic function. Contrary to the standard method of performing logic synthesis, we will assume that each cell can be sized by a continuous, positive real variable S , which increases both the load driving capability of the cell and the area linearly. In other words, the area of a cell is $S \cdot A$ and the delay of a cell is $D = f(C/S)$.

The library analysis will determine a good value for C/S for each cell in the library. Using this value, it determines a constant delay for each gate.

(choice of C/S)

Since the library analysis is not dependent on the actual network being synthesized, library analysis can be performed before beginning the

synthesis process. We will now continue to describe the actual automatic synthesis process, beginning at the with reading the design. The design is expressed in a high level design specification language, for example VHDL or Verilog, and is syntactically parsed and transformed into a logic network representation by a computer program, such as HDL Compiler, available from Synopsys Inc. of Mountain View, California.

(library independent optimization)

Initially the network is library independent and library independent optimizations are performed. Mostly these optimizations change the structure of the network, and the Boolean functions of the cells, without changing the overall function of the network. The types of optimizations that should be performed are behavioral optimizations, such as resource sharing, sequential optimizations such as retiming, algebraic optimizations such as kernel extraction and Boolean optimizations such as redundancy removal. There is a large amount of literature on how each of these classes of optimizations can be performed.

(mapping for delay)

Following the library independent optimizations the network is mapped to a library of cells. This means that the logic functions of the cells are implemented with actual cells from the library. During this process the A large body of literature exists already on the subject of mapping digital networks. The preferred embodiment would use a previously published algorithm such as

(post mapping optimizations)

Due to restrictions in run time, it is impractical to explore the entire design space during the mapping algorithm. Necessarily, the mapping algorithm has to ignore many possible solutions because either they are unlikely candidates or they are very similar (but not

identical) to other, considered solutions. In addition, in the constant delay approach, it is easy to evaluate the impact on timing of synthesis decisions, but it is much harder to evaluate the impact on the total network area. Therefore the mapping algorithm necessarily cannot accurately optimize area.

(pin swapping)

An example of candidate mappings which are not explored during mapping because they are too similar to other mappings follows here: Often gates have several pins which are functionally interchangeable. For instance for a 4 input NAND gate, there are $4 \times 3 \times 2 \times 1$ possible permutations for the 4 input pins. Usually these pins are not equally fast, because of an inherent asymetry of the electronic circuit. Because the differences are small, it is not worthwhile to consider all of these different of different permutations during mapping. It is more efficient to pick one arbitrarily and to select the best permutation of the inputs after mapping.

(structuring - boundary move)

Using constant delay it is considerably easier to predict the effect of a change to the network then with the conventional delay models. This can be used to do timing optimization by means of restructuring after technology mapping has been done. For example, we can use the "boundary move" transformation to reduce the number of levels in the logic in the mapped network. The boundary move transform, illustrated in fig x, reduces the number of levels by bringing connection x forward. To make the change legal it is necessary that gates x, y and z are fanout free. If not, they must be made fanout free by making a copy. In the conventional approach to logic synthesis, copying logic will increase the load on gates x, x, x and therefore increase the delay. To predict if the transformation will improve delay, or hurt

LMAORN.P50

delay, it was necessary to run a complete static timing analysis with accurate delay models. If the change actually worsened the delay, then the change would be undone.

In the constant delay model approach, the effect of this change can be easily predicted. Note that changes in loads do not affect delay. The only change that affects delay, is the change of the fanin of gate xxx. The delay can easily be predicted by simple addition of gate delays.

(area estimation)

To perform area optimization it is necessary to calculate the sizes of the cells. The sizes can be calculated in a straightforward manner from the loads. The loads are calculated by adding the net load and the pin load. The net load consists of the load of the net, which can be estimated using a conventional net load model, plus any other fixed load, such as the load of a primary output. The pin load is not fixed, that is, the load of an input pin depends on the size of the cell. This creates a dependency: To calculate the load of a cell, we need to calculate the size of its fanout cells. Therefore the algorithm starts calculating as far downstream as possible, and traverse the network in a direction opposite to the flow of data. In a combinational network this can be achieved by starting at the primary outputs and traversing the network in a levelized order towards the primary inputs. In a sequential network there may be one or more loops, resulting a cyclic dependency: there is no rightmost cell. In this case the computation can start anywhere in the cycle, and repeats the cycle several times, until the capacitances converge and the error is sufficiently small.

It is possible that this iteration will not converge and that the capacitance will increase in every iteration, by progressively larger amounts. This situation is detected by requiring the increment to

LMAORN.P50

be
smaller than a preset maximum after a fixed number of iterations.
The iteration does not converge if the network is an infeasible
solution: The current network cannot be expected to work at this
speed
because its gain is too small. Changes need to be made to the
network
to increase the gain, which will usually mean increasing the
delay
of the network as well.

After the loads have been calculated the size can be calculated
by
dividing the actual load by the predetermined typical load.
The input capacitance can be
calculated by multiplying the unit gate input capacitance by the
size.
The ratio of these numbers is the size of the gate. The size is
a scale factor, which can be applied to the area of the gate, to
give the area of the sized gate. The area of the network can
be estimated as the sum total of the areas of the sized gates,
plus
the net area as estimated from the total length of all nets.

(net weights)

Various algorithms, such as buffering and placement optimize the
network
by manipulating the loads in the network. Placement optimizes
the net length, which directly related to the net load, and
buffering
reduces the load on a cell by adding extra delay. These
algorithms
can benefit from a more efficient calculation of the effect that
changing
the load of a cell has on sizing. We can do this by calculating
single
parameter per net, called the net weight, which represents the
sensitivity
of the total area of the network with respect to the load on that
net.

This net weight can be calculated in a manner that is very
similar
to the calculation of the pin loads during the area calculation
above.
Starting at the primary inputs, the net weight of the first
(left-most) gate
is equal to its area per unit load. The net weights of the other
cells
can now be calculated with a recurrence relation traversing the

LMAORN.P50

network
from left to right.....

to the calculation of the loads. Starting at the primary inputs the network to 0. Set the net load on the net in question to 1. Perform the iteration and calculate the area as described in the previous section. Since all calculations are linear, the effects simultaneous changes in loads of several nets can be superimposed, that is, added together.

(buffering)

The next step in the synthesis process is the buffering step. The buffering algorithm adds buffers to the network guided by the timing analysis and the area analysis. In the constant delay model, the buffers have a fixed delay, and thus the impact of inserting a buffer is easily determined by subtracting the delay of the buffer from the slack. Thus the effect of adding a buffer on delay is always negative: a buffer always adds delay, never reduces delay. The main effect of adding a buffer is to save area because the source gate can be smaller because the it's load is smaller. The effect on area and gain can be determined by area analysis. Net weights.....

The buffering algorithm works as follows: First it finds locations in the network where a buffer can be added without increasing the network delay. This is done by subtracting the delay of the buffer from the local slack, to give the predicted slack after buffer insertion. If the predicted slack is larger then the network slack, then a buffer can be inserted without increasing the network delay. Next we have to calculate the reduction in load of this net, and check that area that is added by adding the buffer does not exceed the area saved by sizing down the source gate. The area added by inserting the buffer is simply the area of the buffer times its size, where the size is determined by

LMAORN.P50

the load on the buffer divided by the typical load of the buffer. The area saved by inserting the buffer can be calculated by first calculating the change in load due to the insertion of the buffer:

some sinks are removed, the input load of the buffer is added, and the net load estimate may change as a result of the number of fanouts of the net changing. Using the net weight, we can estimate the impact on the network area. If the impact is positive (reduced area) the buffer is inserted. After the buffer is inserted, the capacitances need to be updated in the fanin cone of the buffer, while the net weights need to be updated in the fanout cone of the buffer.

(stretching)

The next step is the process of "Stretching" and "Compressing" the delays of the individual gates to meet the timing constraints. Gates which are on long paths which do not meet the delay constraint are "compressed" until the path does meet the timing constraint. Gates on the short paths which easily meet the timing constraints are "Stretched". Gates with stretched delays require less area for the same load. In this step the delay of the gates is traded against the gain of the gates. When the delay of a gate decreases so does the gain of the gate. It is important that there is enough gain in the network.

Note that this can be done entirely independent of the sizes of the gates. Sizes of gates are not determined until much later in the process.

The stretching algorithm has two phases. In the first phase it will compress the delays of cells on long paths to meet timing constraints. In the second phase it will stretch the delays of the gates on short paths to save area. For the purpose of stretching and compressing registers can usually be considered to be part of a

LMAORN.P50

path
which they originate, but not of a path that they terminate. The stretching algorithm considers the cells on a path by path basis, processing the path with the smallest slack first. The delay of each cell on the path is adjusted by an amount which is equal to the slack divided by the number of stages on the path. After a cell has been adjusted, it becomes "locked" and its delay cannot be changed by the stretching algorithm. Stages which are locked are not counted when calculating the adjustments. For the stretching phase, the algorithm continues as above. In this phase the delays of the cells are increased, not decreased. The path that we work on is not the worst path, but it is the worst path with a slack greater than 0. (All other paths now have a slack of 0).

(Rule based stretching)

(incremental rule based)

(placement)

In our process a conventional placement method is augmented to optimize the area of the placed network. All placement methods known work by gradual refinement of the placement. Periodically, during the placement process, we recalculate the estimated net lengths, using the most recent, accurate placement information. From the net lengths, it calculates the sizes of the cells in the network. (See area estimation). The updated sizes can then be used for further placement and for more accurate net length calculations.

Placement primarily manipulates the lengths of the nets. Using the net weights, the area of the network can efficiently be estimated.

(final or discrete sizing)

LMAORN.P50

Other embodiments will be apparent to those skilled in the art from consideration of the specification and practice of the invention disclosed herein. It is intended that the specification and examples be considered as exemplary only, with a true scope of the invention being indicated by the following claims.

WHAT IS CLAIMED IS:

(CONSTANT DELAY SYNTHESIS)

1. A method for the structuring and mapping of an unmapped digital network comprising the following steps:
 - a) structuring of the digital network, using network slack as an optimization goal, where network slack is calculated assuming that the delay of the cells of the network is constant with respect to load.
 - b) mapping of the cells of the network, using network slack as an optimization goal, where network slack is calculated assuming that the delay of the cells of the network is constant with respect to load.
 - c) estimation of the area of the network based on net load

(typical load)

3. The method of claim 1, the delay of a book is used for the delay of each cell, the delay of the book being chosen by choosing a load size ratio C/S for each book, which is independent of the network.

4. The method of claim 3, where the delay of a book is also determined by the choice of the input transition time, which is independent of the network.

(continuous buffering assumption)

5. The method of claim 3, where a parameter C/S for each book is chosen to have the largest possible value such that a long chain of cells of identical books each cell in the chain having identical value of parameter C/S, said chain cannot have simultaneously improved delay and improved gain by adding a buffer at some point to the same chain, even when the

LMAORN.P50

parameter C/S is chosen optimally after adding the buffer.

(buffering)

6. The method of claim 1, with the additional step of buffer insertion before step c), the buffers being inserted on paths with positive slack, as determined by subtracting the delay of the buffer from the slack of the path.

(more buffering)

7. The method of claim 6, where the buffer is inserted if area is saved.

8. The method of claim 7, where the area savings are estimated using net weights which reflect the change of network area due to sizing as a function of net length.

9. The method of claim 8, where the calculation of the net weights is performed by starting at the primary inputs and traversing the network in the direction of the data flow, while calculating the net weight of a cell by summing over the fanin of the cell the product of the net weight of the fanin cell divided by the gain plus the cells area/load sensitivity.
as $W_i = \sum_j W_j / g_{ij}$

(stretching)

10. The method of claim 1, where step a) is comprising of:
a1) choosing a delay for each book
a2) the delay of the cell assuming the delay of its book
a2) adjusting the delay of each cell based on the slack

11. The method of claim 10, where in step a2) the delay of each cell is adjusted equally among the stages which have the same slack.

12. The method of claim 10, where in step a2) the delay is adjusted on each path, such that the slack of each path becomes 0.

(globally optimal mapping)

13. The method of claim 1, where in step b) the mapping is performed in two steps:
b1) a traversal of the network from primary inputs and registers from left to right, while choosing at each cell the fastest matching books from all available matching

books, using the constant delays of the books and the fastest arrival times of the fanins of the matching book.
b2) a traversal of the network from right to left, while choosing at each cell the fastest matching book from the candidates selected during the previous traversal.

(Area estimation)

14. The method of claim 1, where in step c) consists of the following steps:
c1) estimation of the net length based on the number of fanout cells.
c2) estimation of the capacitive load of the cells using the net length
c3) calculation of the sizes from the capacitive load.
c4) calculation of the network area by summation of the product of the book area times the cell size.

(sizing algorithm)

15. The method of claim 14, where step c2) is performed by starting at the primary outputs and traversing the network in the direction opposite to the data flow, while calculating the load of a cell, by summing over the fanout of the cell, the product of the load of the fanout cell divided by the gain plus the net load of the cell.

(iterate till convergence)

16. The method of claim 15, where the network has loops, the traversal is started at primary outputs and arbitrary cells in the loop, and the calculation is repeated until convergence.

17. The method of claim 16, where the traversal is started at primary outputs and registers.

(Area optimization)

18. The method of claim 1, where steps a) and b) also use network area as an optimization goal, in addition to network delay, the network area being estimated as in step c).

(retiming)

19. The method of claim 1, where the structuring step is preceded

by a retiming step, where registers are moved in the network,
while
preserving the function of the network, and where change to the
network
do not affect the delay of the individual cells.
(I could produce a bunch more of these sort of claims)

(TIMING CLOSURE)

20. A method for the placement and sizing of cells of a mapped
digital network,
the method comprising the steps of:
d) Choosing a target delay for each cell.
e) Computing the network slack using the target delays.
f) Placement of the cells of the network.
g) Sizing of the cells of the network
such that the network meets the network slack as computed by step
b).

(sizing for better placement)

21. The method of claim 20, where step f) is performed in
gradual steps, each step being followed by a sizing step g)

(partitioning)

22. The method of claim 21, where step f) is performed by
repeated partitioning steps,
partitioning the cells in the network into two or more groups,
each group being
assigned to an subdivision of the plane, alternating the
partitioning steps with sizing steps, essentially similar to step
g)

(iterative improvement)

23. The method of claim 21, where step f) is performed by
choosing an arbitrary
initial location in the two dimensional plane for each cell,
the placement being optimized by
repeatedly changing the location of one or two cells at a time,
while
performing a sizing step, essentially similar to step g) after
each location change.

(sizing)

24. The method of claim 5, where step g) is consists of the
following steps:
g1) calculation of the net length based on the available
placement information
g2) calculation of the capacitive load of the cells using the net
length
g3) calculation of the sizes from the capacitive load.

(sizing algorithm)

25. The method of claim 24, where step g2) is performed by starting at the primary outputs and traversing the network in the direction opposite to the data flow, while calculating the load of a cell, by summing over the fanout of the cell, the product of the load of the fanout cell divided by the gain plus the net load of the cell.

(iterate till convergence)

26. The method of claim 10, where the network has loops, the traversal is started at primary outputs and arbitrary cells in the loop, and the calculation is repeated until convergence.

27. The method of claim 26, where the traversal is started at primary outputs and registers.

(cell generation)

28. The method of claim 20, where following step g) the layout of the cells is generated automatically to yield the exact transistor sizes calculated by step g)

(discrete sizing)

29. The method of claim 24, where step g3) consists of selecting the most suitable size from a limited set of available sizes.

(Weighted Placement)

30. A method for the placement of the cells of a digital network, the method comprising the steps, of:
f1) The calculation of net weights that reflect the change of network area due to sizing as a function of net length.
f2) Placement of the cells of the network where the weighted network net length is used as a placement objective, the weighted network net length being the sum of the weighted net lengths of all nets, each net length being multiplied by a weight,

(Calculation of net weights)

31. The method of claim 30, where step f1) is performed by starting at the primary inputs and traversing the network in the direction of the data flow, while calculating the

net
weight of a cell by summing over the fanin of the cell the
product of the net
weight of the fanin cell divided by the gain plus the cells
area/load sensitivity.
as $W_i = \sum_j W_j / g_{ij}$

(iterate till convergence)

32. The method of claim 31, where the network has loops, the
traversal is
started at primary inputs and arbitrary cells in the loop, and
the calculation
is repeated until convergence.

33. The method of claim 32, where the traversal is started at
primary inputs
and registers.

(Weighted Placement for power)

34. A method for the placement of the cells of a digital network,
the
method comprising the steps, of:
f1) The calculation of net weights that reflect the change of
network power due to sizing
as a function of net length.
f2) Placement of the cells of the network
where the weighted network net length is used as
a placement objective, the weighted network net length being the
sum
of the weighted net lengths of all nets, each net length being
multiplied by a weight,

(Calculation of net weights)

35. The method of claim 30, where step f1)
is performed by starting at the primary inputs and traversing the
network in the direction of the data flow, while calculating the
net
weight of a cell by summing over the fanin of the cell the
product of the net
weight of the fanin cell divided by the gain plus the cells
power/load sensitivity.
as $W_i = \sum_j W_j / g_{ij}$

(iterate till convergence)

36. The method of claim 31, where the network has loops, the
traversal is
started at primary inputs and arbitrary cells in the loop, and
the calculation
is repeated until convergence.

37. The method of claim 32, where the traversal is started at
primary inputs

and registers.

38. The method of claim 35, where the power/load sensitivity of a cell is calculated as the product of the switching frequency at that cell times the square of the voltage times the capacitance, where capacitance is calculated as the sum of the net load plus the internal capacitance of the cell, scaled with its size.

ABSTRACT

A method for the design of digital networks consisting of a plurality of cells. The invention uses constant delays during logic synthesis and sizes the cells after placement so as to meet the cycle delay predicted before synthesis. The method chooses a constant delay before logic synthesis and guarantees that it can maintain this delay after placement by means of sizing. Thus it overcomes the unpredictable effects of placement on the cycle delay. The invention also describes methods for choosing the sizes of the gates and a method for inserting buffers.

PROPRIETARY INFORMATION
AND
INVENTIONS AGREEMENT

The following confirms an agreement between me and Synopsys, Inc. (the "Company"), which is a material part of the consideration for my employment by the Company.

1. I recognize that the Company is engaged in a continuous program of research, development and production respecting its business, present and future, including fields generally related to its business and that the Company possesses and continues to possess information that has been created, discovered, developed or otherwise become known to the Company (including, without limitation, information created, discovered or developed by, or made known to, me during the period of or arising out of my employment by the Company) and/or in which property rights have been assigned, licensed or otherwise conveyed to the Company, which information has commercial value in the business in which the Company is engaged. All of the aforementioned information is hereinafter called "Proprietary Information." By way of illustration, but not limitation, Proprietary Information includes trade secrets, processes, data and know-how, computer software, improvements, inventions, works of authorship, techniques, marketing plans, strategies, forecasts and copyrightable material and customer lists.

2. I understand that my employment creates a relationship of confidence and trust between me and the Company with respect to any information:

(i) applicable to the business of the Company; or

(ii) applicable to the business of any client or customer of the Company, which may be known to me by the Company or by any client or customer of the Company, or learned by me during the period of my employment.

3. In consideration of my employment by the Company and the compensation received by me from the Company from time to time, I hereby agree as follows:

A. All Proprietary Information shall be the sole property of the Company and its assigns, and the Company and its assigns shall be sole owner of all patents, copyrights and other rights in connection therewith. I hereby assign to the Company any rights I may have or acquire in such Proprietary Information. At all times, both during my employment by the Company and after its termination, I will keep in confidence and trust all Proprietary Information, and I will not use or disclose any Proprietary Information or anything relating to it without the written consent of the Company, except as may be necessary in the ordinary course of performing my duties to the Company.

B. All documents, records, apparatus, equipment and other physical property, whether or not pertaining to Proprietary Information, furnished to me by the Company or produced by me or others in connection with my employment shall be and remain the sole property of the Company and shall be returned to the Company immediately as and when requested by the Company. Even if the Company does not so request, I shall return and deliver all such property upon termination of my employment by me or the Company for any reason and I will not take with me any such property or any reproduction of such property upon such termination.

C. I will promptly disclose to the Company, or any persons designated by it, all improvements, inventions, works of authorship, processes, techniques, know-how, formulas,

ATTORNEYS AND
CONSULTANTS ONLY

SY000003

data, ideas and other information (including, without limitation, my algorithms or software), whether or not patentable, made or conceived or reduced to practice or learned by me, either alone or jointly with others, during the term of my employment (all said improvements, inventions, works of authorship, processes, techniques, know-how, formulae, data, ideas and other information shall be hereinafter collectively called "Inventions").

D. I agree that all Inventions which I make, conceive, reduce to practice or develop (in whole or in part, either alone or jointly with others) during my employment shall be the sole property of the Company to the maximum extent permitted by Section 2870 of the California Labor Code (hereinafter called "Section 2870"), a copy of which is attached hereto as Exhibit B, and to the extent permitted by law, shall be "works made for hire." The Company shall be the sole owner of all patents, copyrights, trade secret rights, rights with respect to other intellectual property or other rights in connection therewith (including, without limitation, such rights in algorithms or software). I hereby assign to the Company any rights I may have or acquire in such Inventions. I agree to perform, during and after my employment, all acts deemed necessary or desirable by the Company to permit and assist it, at the Company's expense, in obtaining, maintaining and enforcing patents, copyrights, trade secret rights, rights with respect to such Inventions and/or other Inventions I have or may at any time assign to the Company in any and all countries. Such acts may include, but are not limited to, execution of documents and assistance or cooperation in legal proceedings. I hereby irrevocably designate and appoint the Company and its duly authorized officers and agents as my agents and attorneys-in-fact to act for and on my behalf and instead of me, to execute and file any applications or related filings and to do all other lawfully permitted acts to further the prosecution, maintenance and enforcement, issuance of patents, copyrights, trade secret rights, rights with respect to mask works or other rights thereon with the same legal force and effect as if executed by me.

E. As a matter of record, I attach hereto a complete list of all inventions or improvements relevant to the subject matter of my employment by the Company which have been made by me or jointly with others prior to my employment with the Company that I desire to remove from the operation of this Agreement, and I covenant that such list is complete. If no such list is attached to this Agreement, I represent that I have no such inventions and improvements at the time of signing this Agreement.

F. During the term of my employment and for one (1) year thereafter, I will not encourage or solicit any employee of the Company to leave the Company for any reason or devote less than all of any such employee's efforts to the affairs of the Company, provided that the foregoing shall not affect any responsibility I may have as an employee if the Company with respect to the bona fide hiring and firing of Company personnel.

G. I represent that my performance of all the terms of this Agreement will not breach any agreement to keep in confidence proprietary information acquired by me in confidence or in trust prior to my employment by the Company. I have not entered into, and I agree I will not enter into, any agreement, either written or oral, in conflict herewith.

H. I represent that execution of this Agreement, my employment with the Company and my performance of my proposed duties to the Company in the development of its business will not violate any obligations I may have to my former employer.

I. This Agreement does not require assignment of an invention which an employee cannot be obligated to assign under Section 2870. However, I will disclose any Inventions as required by Section 3(c) hereof regardless of whether I believe the Invention is protected by Section 2870, in order to permit the Company to engage in a review process to determine such issues as may arise. Such disclosure shall be received in confidence by the Company.

SY000004

**ATTORNEYS AND
CONSULTANTS ONLY**

4. This Agreement shall be effective as of the first day of my employment by the Company: June 26 1995 (date).

5. This Agreement shall be binding upon me, my heirs, executors, assigns and administrators and shall inure to the benefit of the Company, its successors and assigns.

Dated: 5/17/95



Signature

Lukas van Ginneken

(Print or Type Name)

Accepted and agreed to:

Synopsys, Inc.

By  _____

ATTORNEYS AND
CONSULTANTS ONLY

SY000005

EXHIBIT A
TO
SYNOPSIS, INC. PROPRIETARY INFORMATION
AND INVENTIONS AGREEMENT

1. The following is a complete list of all inventions or improvements relevant to the subject matter of my employment by Synopsis, Inc. (the "Company") that have been made or conceived or first reduced to practice by me alone or jointly with others prior to my employment by the Company that I desire to remove from the operation of the Company's Proprietary Information and Inventions Agreement.

_____ No inventions or improvements

_____ See below

X Additional sheets attached

2. I propose to bring to my employment the following materials and documents of a former employer:

X No materials or documents

_____ See below



Signature

Lukas van Glnneken,

Print or Type Name

ATTORNEYS AND
CONSULTANTS ONLY

SY000006

EXHIBIT B
TO
SYNOPSYS, INC. PROPRIETARY INFORMATION
AND INVENTIONS AGREEMENT

Section 2870. Application of provision providing that employee shall assign or offer to assign rights in inventions to employer.

(a) Any provision in an employment agreement which provides that an employee shall assign, or offer to assign, any of his or her rights in an invention to her or her employer shall not apply to an invention that the employee entirely on his or her own time without using the employer's equipment, supplies, facilities or trade secret information except for those inventions that either:

(1) Relate at the time of conception or reduction to practice of the invention to the employer's business, or actual or demonstrably anticipated research or development by the employer;

(2) result from any work performed by the employee for the employer.

(b) To the extent a provision in an employment agreement purports to require an employee to assign an invention otherwise excluded from being required to be assigned under subdivision (a), the provision is against the public policy of this state and is unenforceable.

ATTORNEYS AND
CONSULTANTS ONLY

SY000007

The following is a complete list of all inventions or improvements relevant to the subject matter of my employment by Synopsys, Inc. (the Company) that have been made or conceived or first reduced to practice by me alone or jointly with others prior to my employment by the Company that I desire to remove from the operation of the Company's Proprietary Information and Inventions Agreement.

[0] "Efficient orthonormality testing for synthesis with pass transistor selectors" by M. R. C. M. Berkelaar and -, accepted at the International Workshop on Logic Synthesis, June 1995.

[1] "Timing Verification and Optimization for the PowerPC Processor Family", by R.E. Mains, T. A. Mosher, - and R.F. Damiano, in: Proc. Int. Conf. on Computer Design, pp.390-393, Boston, Oct. 10-12, 1994.

[2] "In the driver's seat of BooleDozer" by D. Brand and R.F. Damiano, -, A. D. Drumm, in: Proc. Int. Conf. on Computer Design, pp. 518-521, Boston, Oct. 10-12, 1994.

[3] "Grammar-based optimization of synthesis scenarios" by A. Kuehlmann and -, in: Proc. Int. Conf. on Computer Design, pp. 20-25 Boston, Oct. 10-12, 1994.

[4] "Tuning of logic synthesis scenarios" by - and A. Kuehlmann, Workshop notes of the Int. Workshop on logic synthesis, paper P7c, Tahoe City, May 23-26, 1993.

[5] "Fanin ordering in multi-slot timing" by -, Proc. Int. Conf. on Computer Design, pp. 44-47, Cambridge, Oct. 11-14, 1992.

[6] "The complexity of adaptive annealing" by R. H. J. M. Otten and -, Proc. Int. Conf. on Computer Design, pp. 404-407, Cambridge, Sept. 17-19, 1990.

[7] "Buffer placement in distributed RC-tree networks for minimal Elmore delay" by -, Proc. Int. Symp. on Circuits and Systems, pp. 865-868, New Orleans, May 2-5, 1990.

[8] "Optimal slicing of plane point placements" by - and R. H. J. M. Otten, Proc. European Design Automation Conf. pp. 322-236, Glasgow, March 12-15, 1990.

[9] The annealing algorithm by R. H. J. M. Otten and -, ISBN 07923-9022-9, Boston:Kluwer, 1989.

[10] The predictor-adaptor paradigm - automation of custom layout by flexible design by -, Ph.D. thesis, ISBN 90-9002703-3, Eindhoven, 1989.

[11] "Doubly folded transistor matrix layout" by - and J. T. J. van Eijndhoven, A. H. C. M. Brouwers, Digest Int. Conf. on Computer Aided Design, Santa Clara, Nov. 7-10, 1988.

[12] "Stop criteria in simulated annealing" R. H. J. M. Otten and -, Proc.

Int. Conf. on Computer Design, pp.549-552, Port Chester, Oct. 3-5, 1988.

[13] "An inner loop criterion for simulated annealing" by - and R.H.J.M. Otten, Physics letters A, 130:429-435, 1988.

[14] "Soft Macro Cell generation by two dimensional folding" by - and J. T. J. van Eijndhoven, P. R. M. van Teeffelen, T. J. Deckers, Proc. Int. Symp. on Circuits and Systems, pp. 727-730, Espoo, June 1988.

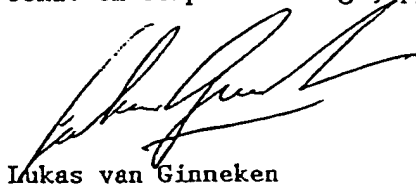
[15] "Gridless routing of general floor plans" by - and J. A. G. Jess, Digest Int. Conf. on Computer Aided Design, pp. 30-33, Santa Clara Nov. 9-12, 1987.

[16] "Wire planning for stackable designs", by R. K. Brayton, C. L. Chen, J. A. G. Jess, R. H. J. M. Otten and -, Proc. Int. Symp. on VLSI technology, pp.269-273, Taipeh, May 13-15, 1987.

[17] "Global wiring for custom layout design" by - and R. H. J. M. Otten, Proc. Int. Symp. on Circuits and Systems. pp.207-208, Kyoto, June 5-7, 1985.

[18] "Floor plan design using simulated annealing" by R. H. J. M. Otten and -, Digest Int. Conf. on Computer Aided Design, pp. 96-98, Santa Clara, Nov. 1984.

[19] "Stepwise layout refinement" by - and R. H. J. M. Otten, Proc. Int. Conf. on Computer Design, pp. 30-36, Port Chester, Oct.8-11, 1984.



Lukas van Ginneken

Synopsys Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 2</u></p> <p>The major problem with the <i>conventional approach</i> is that the net length and hence the cell delay is not known <i>until after placement</i>.</p>	<p><u>'446 PATENT AT 1:46-47</u></p> <p>Thus, under the <i>conventional design approach</i>, timing closure is not certain <i>until after placement</i>.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 2-3</u></p> <p>Before placement, <i>net length</i> must be <i>estimated</i>. This is usually done with <i>an estimation function or table which gives the load of a net based on its fanout</i>. Experience has shown that it is very difficult to <i>estimate</i> the length of the nets <i>accurately</i>.</p>	<p><u>'446 PATENT AT 1:37-40</u></p> <p>While <i>net lengths</i> have been <i>estimated</i> prior to placement by use of <i>an estimation function or table which gives the load value of a net based on the number of fanout gates</i>, this <i>estimation function</i> is usually <i>inaccurate</i>.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 3</u></p> <p>The result is unpleasant surprises <i>after placement</i> step 105. <i>Some nets turn out to be longer than expected</i>, and because of the <i>longer delays</i>, the <i>timing constraints</i> are not met. <i>Timing closure is not certain until after</i> step 105.</p>	<p><u>'446 PATENT AT 1:41:46</u></p> <p>This difficulty in accurately predicting net lengths leads to unpredictable delay effects <i>after cell placement</i> occurs. For example, <i>some nets turn out to be longer in length than expected</i>. These longer nets cause <i>longer delays</i> which prevent satisfaction of <i>timing constraints</i> in the digital circuit. Thus, under the conventional design approach, <i>timing closure is not certain until after</i> placement.</p>

¹ Note that page numbers do not appear on the original Draft Patent Application, but have been added for convenience. No other changes were made to the Draft Patent Application.

Synopsys Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSYS DRAFT PAT. APP. AT 3</u></p> <p>If <i>timing closure</i> is not <i>achieved</i> the options the <i>designer</i> has are <i>expensive</i> and unreliable. He may choose to <i>fix the design manually, which is difficult and time consuming, because the automatically optimized network</i> is hard to <i>understand</i>. He may choose to <i>change</i> his <i>HDL specification and repeat the synthesis process</i>. Again <i>timing closure will not be certain until after placement</i>, which means that the entire <i>process</i> needs to be traversed <i>before the designer</i> knows <i>if</i> his <i>HDL changes were successful</i>.</p>	<p><u>'446 PATENT AT 1:48-60</u></p> <p>Failure to <i>achieve timing closure</i> after placement leads to additional <i>expenses</i> and other problems for the <i>designer</i>. To correct for failure to achieve timing closure, the <i>designer</i> has the option of <i>fixing the design manually, which is difficult and time consuming because the automatically optimized digital network</i> is not easy to <i>understand</i>. As a second option, the designer may <i>change</i> the Hardware Description Language (<i>HDL</i>) <i>specification and repeat the design process</i>. However, <i>timing closure will again not be certain until after placement</i>. Thus, the design <i>process</i> must again be repeated <i>before the designer</i> can determine <i>if</i> the <i>HDL</i> specification <i>changes were successful</i> in enabling timing closure.</p>
<p><u>SYNOPSYS DRAFT PAT. APP. AT 3</u></p> <p>A common method of dealing with <i>inaccurate net load estimates</i> is to use <i>net load estimates</i> which are <i>considerably larger than accurate estimates</i>. This <i>causes the sizes of the cells to be considerably larger than necessary</i> but reduces the <i>probability of not meeting the timing constraints after placement</i>. Clearly using cells with sizes <i>which are larger than necessary</i> is <i>wasteful in both silicon area and power consumption</i>. The <i>chips</i> thus synthesized will be <i>larger, cost more to produce and use more electrical power than necessary</i>.</p>	<p><u>'446 PATENT AT 1:61-2:3</u></p> <p>A common method for dealing with <i>inaccurate net load estimates</i> is by <i>estimating the net load</i> at a <i>considerably larger value than typically estimated</i>. Although this method increases the <i>probability of meeting timing constraints after placement</i>, it <i>causes the sizes of the gates to be considerably larger than necessary</i>. Gates which are <i>larger than the necessary size</i> are <i>wasteful in both silicon area and power consumption</i>. This leads to <i>chips</i> which are <i>larger, more expensive to produce, and use more electrical power than necessary</i>.</p>
<p><u>SYNOPSYS DRAFT PAT. APP. AT 3</u></p> <p>A second <i>problem with the conventional approach</i> is that the effect of synthesis decisions is hard to calculate. <i>Performing timing analysis during optimization is very time consuming, and accounts for most of the run time of conventional synthesis systems</i>.</p>	<p><u>'446 PATENT AT 2:4-9</u></p> <p>Another <i>problem with the conventional circuit design approach</i> concerns the timing analysis required <i>during optimization</i> and during placement. The <i>timing analysis performed</i> throughout the conventional circuit design process <i>is very time consuming, and accounts for most of the run time of a conventional circuit design system</i>.</p>

Synopsis Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 4</u></p> <p>In step 105 the placement program will <i>modify the net lengths. Depending on which location was chosen for each cell, the length of each net can be different. As the length differs, the capacitive load of the net changes</i>, and as a result, the delay of the cell driving the net changes. <i>Therefore the delays which were carefully optimized during the logic synthesis, are very different after placement</i>, and the <i>optimization</i> of the <i>network</i> is not very good.</p>	<p><u>'446 PATENT AT 2:12-19</u></p> <p><i>Depending on the location chosen for each gate, each net length may be modified. As each net length is modified, the capacitive load of the net will change. Therefore, the delays, which were carefully optimized during the logic design, are very different in value after cell placement</i>, thereby contributing to poor <i>network optimization</i>.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 4</u></p> <p><i>Much of the progress in the state of the art can be characterized as increased integration</i>. This is represented in figure 1 as various feedback paths, which repeat and alternate steps. The general direction has been towards programs which do structuring, mapping, sizing and placement simultaneously. <i>It has led to increasingly complex software systems which are slow and difficult to design and maintain</i>.</p>	<p><u>'446 PATENT AT 2:20-23</u></p> <p>Additionally, <i>much of the progress in the state of the art for digital circuit design can be characterized as increased integration which has led to increasingly complex software systems which are slow, and difficult to design and maintain</i>.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 4</u></p> <p><i>Iterating between placement and sizing</i> has been especially hard to execute because placement programs are not sold by the same design automation software vendors as <i>logic synthesis programs</i>. Also they are not run by the same users: the logic synthesis program is often run by the <i>designer, who also wrote the HDL specification</i>. The <i>placement program</i> is often run by the <i>silicon chip manufacturer, after the design is considered complete</i>.</p>	<p><u>'446 PATENT AT 2:24-30</u></p> <p>A further disadvantage with conventional design approaches is in the difficulty of <i>iterating between placement and sizing</i>, since the <i>logic synthesis program</i> is often operated by the <i>logic designer who also wrote the HDL specification</i>, but the <i>placement program</i> is often operated by the <i>silicon chip manufacturer, after the design is complete</i>.</p>

Synopsis Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 6</u></p> <p>The <i>present invention maintains timing closure</i> after it has been achieved by <i>adjusting the size of the cell during or after placement</i>. The <i>adjustments compensate for the fact that the placement algorithm can assign different net lengths to different nets and that these lengths are difficult to predict before placement</i>.</p>	<p><u>'446 PATENT AT 16:23-29</u></p> <p>According to the <i>present invention, timing closure is maintained</i> after placement occurs of cells 836. To <i>maintain timing closure</i>, the size of a particular gate may be <i>adjusted during or after placement</i>. This <i>adjustment compensates for the fact that placement algorithm may assign different net lengths to different nets, and that these different net lengths are difficult to predict prior to the placement step</i>.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 8</u></p> <p>Wherever possible, the same <i>reference numbers</i> will be used throughout the <i>drawings to refer to the same or like parts</i>.</p>	<p><u>'446 PATENT AT 4:59-63</u></p> <p><i>Referring in detail now to the drawings wherein similar parts or steps of the present invention are identified by like reference numerals</i>, there is seen in FIG. 1 a schematic diagram of a host computer system 100 which is capable of implementing the present invention.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 9</u></p> <p>The cells <i>can be combinational "gates"</i> 207, 208, 209, <i>whose function is represented as an expression in the Boolean algebra, using AND, OR and NOT operators</i>, or the cells <i>can be registers</i> 205, 206.</p>	<p><u>'446 PATENT AT 5:13-17</u></p> <p>The gates <i>can be combinational gates whose function is represented as Boolean expression</i> based on, for example, the <i>operators AND, OR and NOT</i>. The gates <i>can also be registers</i>.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 9</u></p> <p><i>Each cell (e.g., 208) has one or more inputs 212, 213, and a single output 214.</i></p>	<p><u>'446 PATENT AT 5:18-19</u></p> <p><i>Each gate (e.g., gate j) has one or more input 155 and a single output 160.</i></p>

Synopsis Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 9</u></p> <p><i>Cells whose inputs are connected to the output of a cell are called the fanin of the latter cell. Cells whose inputs are connected to the output of a cell are called the fanout of the latter cell.</i></p>	<p><u>'446 PATENT AT 5:26-32</u></p> <p><i>Gates whose outputs are connected to the inputs of a gate are collectively called the "fanin" of the latter gate. Thus, the gate k is in the fanin of the gate i. Gates whose inputs are connected to the output of a gate are collectively called the "fanout" of the latter gate. Thus, the gate j is in the fanout of the gate i.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 9</u></p> <p><i>The digital network performs a logic "function" by processing digital binary input data in a number of cycles. The input data is presented to the network on its *primary inputs* 201, 202, and the result of the computation of the network function is presented at the *primary outputs* 203, 204, of the network. The computation of the function takes one or more cycles. During each cycle the gate functions are calculated. The results are stored in the registers for use in the next cycle.</i></p>	<p><u>'446 PATENT AT 5:33-41</u></p> <p><i>The digital circuit 150 performs a logic function by processing digital binary input data in a number of cycles. The input data is presented to the digital circuit 150 at the primary inputs 170, and the result of the computation of the digital circuit function is presented at the primary outputs 175. Typically, the computation of the digital circuit function requires one or more cycles. During each cycle, the gate functions are calculated, and the calculation results are stored in registers for use in the next cycle.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 10</u></p> <p><i>The "arrival time" of the data at a gate is computed by taking the maximum arrival time of its fanin cells each increased by the delay from the input pin to the output pin.</i></p>	<p><u>'446 PATENT AT 9:55-58</u></p> <p><i>(An arrival time of the data at a gate is computed by taking the maximum arrival time of the fanin gates plus the delay measured from the input pin to the output pin of the gate).</i></p>

Synopsis Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 11</u></p> <p>The difference between the required time and the arrival time is the <i>*slack*</i>. If the arrival time is smaller than the required time, <i>the timing constraints are met</i>, and the <i>slack is positive</i>. If the arrival time is larger than the required time, the timing constraints are not met, and the slack is negative. The arrival time and required time may be different depending on whether the data is zero (0) or one (1). There also may be multiple arrival times and multiple required times to model a variety of timing constraints. <i>All slacks can be summarized as a single worst slack number, called the *network slack*. Timing closure is achieved if the network slack is non-negative.</i></p>	<p><u>'446 PATENT AT 13:27-34</u></p> <p>This determination is made by subtracting the delay of the buffer from the "local <i>slack</i>", to give the value of the predicted slack after buffer insertion. <i>Slack is zero or positive if the timing constraints are met</i>. In addition, <i>all slacks in the circuit can be summarized by the "network slack" which is the single "worst" slack number. If the network slack is non-negative, then the timing closure is achieved.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 11</u></p> <p>It is important to note that the dependency on size and load can be captured as the dependency on a single parameter C/S, and <i>the delay D is non-negative and monotonically increasing with C/S.</i></p>	<p><u>'446 PATENT AT 6:38-43</u></p> <p>The delay D of a gate can be approximated by equation (1):</p> $D=f(C/S) \quad (1)$ <p><i>The delay D is non-negative and increases as the C/S value increases.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 11</u></p> <p><i>The delay may be different for different inputs of the gate and it may be different for the falling and the rising transition.</i></p>	<p><u>'446 PATENT AT 6:58-61</u></p> <p><i>The delay D value may also be different for different inputs of the gate and it may also be different for the falling transition and rising transition of a signal propagating through the gate.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 12</u></p> <p><i>The library analysis will determine a good value for C/S for each cell in the library.</i></p>	<p><u>'446 PATENT AT 6:63-65</u></p> <p><i>The library analysis will determine a "good" value for C/S for each gate in the library based on gain considerations.</i></p>

Synopsis Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 13</u></p> <p>Mostly these optimizations <i>change the structure of the network, and the Boolean functions of the cells, without changing the overall function of the network</i>. The types of <i>optimizations</i> that should be performed are <i>behavioral optimization such as resource sharing, sequential optimizations such as retiming, algebraic optimizations such as kernel extraction and Boolean optimizations such as redundancy removal</i>. There is a large amount of literature on how each of these <i>classes of optimizations</i> can be performed.</p>	<p><u>'446 PATENT AT 9:13-22</u></p> <p>During this step, <i>the structure of the circuit and the Boolean functions of the gates are changed</i> to reduce the total number of connections, <i>without changing the overall function of the circuit</i>. Structural <i>optimizations</i> can include <i>behavioral optimizations (such as resource sharing), sequential optimizations (such as retiming), algebraic optimizations (such as kernel extraction), and Boolean optimizations (such as redundancy removal)</i>. The <i>classes of optimizations</i> above are well known to those skilled in the art.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 13</u></p> <p>Following the library independent optimizations, the network <i>is mapped to a library of cells</i>. This means that <i>the logic functions of the cells are implemented with actual cells from the library</i>.</p>	<p><u>'446 PATENT AT 9:25-27</u></p> <p>In step 210 (FIG. 4), the circuit <i>is mapped to a library 209 of cells</i>. Thus, <i>the logic functions of the circuit gates are implemented with actual cells from the library 209</i>.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 14</u></p> <p>For example, we can <i>use the "boundary move" transformation to reduce the number of levels in the logic in the mapped network</i>. The boundary move <i>transform</i>, illustrated in fig x, <i>reduces the number of levels by bringing connection x forward</i>.</p>	<p><u>'446 PATENT AT 10:45-49</u></p> <p>A local <i>transformation</i> is then <i>used to reduce the number of levels in the logic in the gate chain circuit 550</i>. The result of the <i>transformation</i> is shown as gate chain circuit 550' in FIG. 7B. <i>The number of levels in the logic is reduced by bringing the gate 555 forward</i>.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 14</u></p> <p>To make the change legal <i>it is necessary that gates x, y and z are fanout free. If not, they must be made fanout free</i> by making a copy.</p>	<p><u>'446 PATENT AT 10:59-62</u></p> <p>In order for the transformation shown in FIG. 7B to be valid, <i>it is necessary that gates 555, 560, and 565 are fanout free. If the gates 555, 560, and 565 are not fanout free, then they are made fanout free through copying logic</i>.</p>

Synopsis Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 14-15</u></p> <p>In the <i>conventional</i> approach to logic synthesis, <i>copying logic will increase the load on gates</i> x, x, x and therefore increase the delay. <i>To predict if the transformation will improve delay, or hurt delay, it was necessary to run a complete static timing analysis with accurate delay models. If the change actually worsened the delay, then the change would be undone.</i></p>	<p><u>'446 PATENT AT 11:4-13</u></p> <p>Under <i>conventional logic</i> design, <i>copying logic will increase the load on the gates</i> whose outputs are connected to lines 575, 580, 585, and 590. In the example of FIG. 7B, the copying logic 555' <i>increases the load on the gates</i> whose outputs are connected to lines 575 and 580. <i>To predict whether or not the transformation improved delay, it is necessary to run a complete static timing analysis with accurate delay models. If the transformation (from circuit 550 to 550') were actually harmful to delay, then the transformation would have to be undone.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 15</u></p> <p><i>In the constant delay model approach, the effect of this change can be easily predicted. Note that changes in loads do not affect delay. The only change that affects delay, is the change of the fanin of gate xxx. The delay can easily be predicted by simple addition of gate delays.</i></p>	<p><u>'446 PATENT AT 10:49-58</u></p> <p><i>In the constant delay model approach, the effect of this transformation can be easily predicted. Changes in the gate loads do not affect delay, since delay is maintained as constant while gate size will be adjusted (during or after placement) to compensate for the load change. The only change which affects delay (of the gate chain circuit 550) is the change of the fanin of gate 555. This delay change can be predicted by simple addition of gate delays provided by the fanins connected at lines 590, 575, and 580 (see gate chain circuit 550' in FIG. 7B).</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 15</u></p> <p>The <i>net load</i> consists of the <i>load</i> of the <i>net</i>, which <i>can be estimated using a conventional net load model, plus any other fixed load, such as the load of a primary output.</i></p>	<p><u>'446 PATENT AT 11:26-30</u></p> <p>The parameter w represents the <i>net</i> (wire) <i>load</i> for a given gate i (wherein the <i>net load can be estimated using a conventional net load model</i> such as the above-mentioned fanout-based model) <i>plus any other fixed load such as the load of the primary output</i> of the circuit implementation.</p>

Synopsis Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 15</u></p> <p>In <i>a combinational network</i> this can be achieved by starting <i>at the primary outputs and traversing the network in a leveled order towards the primary inputs.</i></p>	<p><u>'446 PATENT AT 11:48-52</u></p> <p>If the digital circuit is <i>a combinational network</i> (see, e.g. circuit 150 in FIG. 2), then gate load calculation initiates <i>at the primary outputs 175 and traverses the circuit in a leveled order toward the primary inputs 170.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 15-16</u></p> <p>In <i>a sequential network there may be one or more loops, resulting a cyclic dependency: there is no rightmost cell.</i> In this case the computation <i>can start anywhere in the cycle, and repeats the cycle several times, until the capacitances converge</i> and the error is <i>sufficiently small.</i> It is possible that this iteration will <i>not converge and</i> that the <i>capacitance will increase</i> in every iteration, <i>by progressively larger amounts.</i> This situation is <i>detected</i> by requiring the increment to be smaller than <i>a preset maximum after a fixed number of</i> iterations. The iteration does <i>not converge</i> if the network is <i>an infeasible solution:</i> The current network cannot be <i>expected to work at this speed because its gain is too small. Changes</i> need to be made to the network to <i>increase the gain, which will usually mean increasing the delay</i> of the network as well.</p>	<p><u>'446 PATENT AT 11:53-12:4</u></p> <p>If the digital circuit is <i>a sequential network</i> (see, e.g., circuit 180 of FIG. 3), then <i>there may be one or more loops</i> (e.g., loop 182) which <i>result in a cyclic dependency</i> (i.e., <i>there is no "rightmost" gate</i>). Gate load calculation <i>can start anywhere in the cycle, and</i> calculation in <i>the cycle</i> is performed <i>several times until the load capacitance values converge</i> or have <i>sufficiently small</i> differences. However, a condition may exist when the load <i>capacitance</i> values do <i>not converge and increase by progressively larger amounts</i> every cycle calculation. This increase in load capacitance values can be <i>detected</i> if the calculated load values exceed <i>a preset maximum value after a fixed number of</i> cycle calculations. When the calculated load values do <i>not converge</i>, then the particular circuit 180 has <i>an infeasible solution</i>, which indicates that the digital circuit is <i>not expected to work at the set speed because the circuit gain is too small. Changes</i> are required to <i>increase the circuit gain</i>, and these changes <i>will usually</i> lead to an increase in circuit <i>delay.</i></p>

Synopsis Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 16</u></p> <p>After the loads have been calculated the <i>size</i> can be calculated <i>by dividing the actual load by the predetermined typical load</i>. The input capacitance can be calculated by multiplying the unit gate input capacitance by the size. The ratio of these numbers is the <i>size</i> of the <i>gate</i>. <i>The size is a scale factor, which</i> can be applied to the area of <i>the gate</i>, to give <i>the area of the sized gate</i>. <i>The area of the network can be estimated as the sum total of the areas of the sized gates, plus the net area as estimated from the total length of all nets.</i></p>	<p><u>'446 PATENT AT 12:5-20</u></p> <p>In the above example, the <i>size S</i> of a gate <i>i</i> is determined <i>by dividing the actual load C_i by the predetermined typical load C/S</i> of the gate <i>i</i>. The size <i>S</i> is a scale factor which is applied to all transistor channel widths of a gate in order to determine the area of the "<i>sized gate</i>". <i>The size S is also a scale factor which</i> is used to scale <i>the gate's</i> output load driving capability and its input pin loads. <i>The area of the sized gate</i> is determined by equation (5).</p> <p>area of sized gate = $S \times (\text{area of gate})$ (5)</p> <p><i>The area of the mapped digital circuit can be estimated based on the sum of the total areas of the sized gates plus the net area</i> (which is estimated from the total length of all nets in the circuit).</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 16-17</u></p> <p>We can do this by <i>calculating</i> single parameter per net, called the <i>net weight</i>, which <i>represents the sensitivity of the total area of the network with respect to the load on that net</i>. This <i>net weight</i> can be calculated in a manner that is very similar to the calculation of the pin loads during the area calculation above. Starting at <i>the primary inputs, the net weight of the first (left-most) gate is equal to its area per unit load</i>. <i>The net weights of the other cells can now be calculated</i> with a recurrence relation traversing the network from left to right.</p>	<p><u>'446 PATENT AT 12:22-30</u></p> <p>Thus, the following discussion now turns to the <i>calculation of "net weights."</i> The <i>net weight represents the sensitivity of the total area of a digital circuit with respect to the load of a particular net</i>. As an example, <i>the net weight of a given gate, which is immediately coupled to the primary inputs of a digital circuit, is equal to its area per unit load</i>. Using equation (6), <i>the net weight of the other gates in the digital circuit are then calculated</i> in a leveled order towards the primary outputs of the digital circuit.</p>

Synopsis Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 17</u></p> <p>The <i>buffering</i> algorithm works as follows: First it finds <i>locations in the network where a buffer can be added</i> without increasing the network delay. This is done <i>by subtracting the delay of the buffer from the local slack, to give the predicted slack after buffer insertion. If the predicted slack is larger then the network slack, then a buffer can be inserted without increasing the network delay.</i></p>	<p><u>'446 PATENT AT 13:23-37</u></p> <p>The <i>buffering</i> step of 215 (FIG. 4) is discussed in further detail with reference to FIG. 8. In step 650, <i>locations in the circuit are determined where a buffer can be added</i> so that buffer insertion will still permit timing constraints to be met. This determination is made <i>by subtracting the delay of the buffer from the "local slack", to give the value of the predicted slack after buffer insertion.</i> Slack is zero or positive if the timing constraints are met. In addition, all slacks in the circuit can be summarized by the "network slack" which is the single "worst" slack number. If the network slack is non-negative, then timing closure is achieved. <i>If the predicted slack calculated in step 650 is larger than the network slack, then it is possible to insert a buffer without increasing the circuit delay.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 17-18</u></p> <p>Next we have to calculate the reduction in load of this net, and check that <i>area that is added by adding the buffer does not exceed the area saved by sizing down the source gate.</i> The <i>area added by inserting the buffer</i> is simply <i>the area of the buffer times its size, where the size is determined by the load on the buffer divided by the typical load of the buffer.</i> The <i>area saved by inserting the buffer</i> can be calculated by first <i>calculating the change in load due to the insertion of the buffer: some sinks are removed, the input load of the buffer is added, and the net load estimate may change as a result of the number of fanouts of the net changing.</i></p>	<p><u>'446 PATENT AT 13:37-48</u></p> <p>In step 655, it is determined whether the <i>added area due to buffer insertion does not exceed the area saved by sizing down the source gate.</i> The <i>added area (by inserting the buffer)</i> is equal to <i>the area of the buffer multiplied by the buffer size, wherein the buffer size is determined by the buffer load C divided by the typical load C/S on the buffer.</i> The <i>area saved by sizing down the source gate</i> is determined by first <i>calculating the change in net load due to the buffer insertion.</i> This <i>net load change</i> is due to the following: (1) <i>some sinks (which sink currents) are removed,</i> (2) <i>the input load of the buffer is added, and</i> (3) <i>the number of fanouts of the gate may change.</i></p>

Synopsys Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 18</u></p> <p><i>After the buffer is inserted, the capacitances need to be updated in the fanin cone of the buffer, while the net weights need to be updated in the fanout cone of the buffer.</i></p>	<p><u>'446 PATENT AT 13:53-57</u></p> <p><i>After the buffer has been inserted, then in step 670 the capacitance values need to be updated in the fanin cone of the buffer, while the net weights need to be updated in the fanout cone of the buffer.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 18</u></p> <p>The next step is the process of "<i>Stretching</i>" and "<i>Compressing</i>" the delays of the individual gates to meet the timing constraints. Gates which are on long paths which do not meet the delay constraint are "<i>compressed</i>" until the path does meet the timing constraint. Gates on the short paths which easily meet the timing constraints are "<i>Stretched</i>". Gates with stretched delays require less area for the same load. In this step the delay of the gates is traded against the gain of the gates. When the delay of a gate decreases so does the gain of the gate. It is important that there is enough gain in the network.</p>	<p><u>'446 PATENT AT 14:20-36</u></p> <p>Prior to cell placement, the delays of the individual gates may be stretched or compressed to meet the delay constraints, as shown in step 220 of FIG. 4. As shown in FIG. 9A, by compressing (decreasing) the delay of a given gate, the gate gain decreases. Gates which are on long paths not meeting the delay constraints are compressed (in delay) until the long paths meet the delay constraints. The delay of the gates (or gate) may be decreased as long as the minimum required gain requirements are met. By stretching (increasing) the delay of a given gate, the gate gain increases (see FIG. 9A). Gates on short paths which easily meet the delay constraints are stretched (in delay), since gates with stretched delays require less area for the same load. The delay of the gates (or gate) in a path are stretched to the extent that timing constraints for the digital circuit are still met.</p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 18-19</u></p> <p><i>For the purpose of stretching and compressing registers can usually be considered to be part of a path which they originate, but not of a path that they terminate.</i></p>	<p><u>'446 PATENT AT 15:48-51</u></p> <p><i>For the purpose of stretching and compressing, registers in the circuit are preferably considered as part of a path from which they originate, but not part of the path from which they terminate.</i></p>
<p><u>SYNOPSIS DRAFT PAT. APP. AT 19</u></p> <p>The stretching algorithm considers the cells on a path by path basis, processing the path with the smallest slack first.</p>	<p><u>'446 PATENT AT 15:9-10</u></p> <p>The invention operates on a path-by-path basis whereby the most critical path in a digital circuit 750 is evaluated first.</p>

Synopsis Draft Patent Application ¹	Specification Of '446 Patent (Also Contained in '438 Patent)
<p><u>SYNOPSIS DRAFT PAT. APP. AT 19</u></p> <p>After a cell <i>has been adjusted</i>, it becomes "<i>locked</i>" and its <i>delay cannot be</i> changed by the <i>stretching</i> algorithm.</p>	<p><u>'446 PATENT AT 15:21-25</u></p> <p>After the gate 754 <i>has been adjusted</i> to meet the Path 2 timing constraints, <i>it becomes "locked,"</i> whereby the gate 754 delay will <i>not be</i> adjusted further for the remainder of the compression and <i>stretching</i> step.</p>

DECLARATION OF ROBERT DAMIANO

I, Robert Damiano, declare as follows:

1. The following statements are based on my personal knowledge. If called upon to testify, I could and would competently testify as to the matters set forth herein.
2. I am an employee of Synopsys, Inc. My present position is Vice-President of the Advanced Technology Group. In September 1996, my position at Synopsys was Director in the Advanced Technology Group. Lukas van Ginneken, who was also employed at Synopsys during that time, worked on one of my projects.
3. On or about September 9, 1996, I received an email from Lukas van Ginneken that included a draft patent application. A true and correct copy of such email is attached hereto as Exhibit A.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of Declarant:

Robert Damiano

Declarant's Signature:

Robert Damiano

Date:

28 January 2005

Exhibit A

[Document Follows]

To: robertd@synopsys.com
From: Lukas van Ginneken <lukas@synopsys.com>
Subject: patent
Date: 1996-09-09 22:07:54 GMT

SYNOPSYS CONFIDENTIAL

APPLICATION FOR UNITED STATES PATENT

in the name of

LUKAS PAUL PIETER PEPIJN VAN GINNEKEN

of

SYNOPSYS, INC.

for

METHOD FOR ACHIEVING TIMING CLOSURE OF DIGITAL NETWORKS

AND

METHOD FOR AREA OPTIMIZATION OF DIGITAL NETWORKS UNDER TIMING CLOSURE

class 364/489

BACKGROUND OF THE INVENTION

This application relates to a method for achieving timing closure of digital networks consisting of structuring and mapping and a method for area optimization of digital networks using placement and sizing, while maintaining timing closure.

(Prior Art)

Figure 1 shows the conventional approach to digital network synthesis.

Digital network synthesis is a process in which computer programs optimizes digital networks. At the beginning of the synthesis process, a human designer 110 specifies a design 111 at a high level of abstraction using a high level design specification language, such as Verilog or VHDL.

In step 101 the abstract network specification of the design is transformed into an unmapped digital network representation in memory by a computer program, such as HDL Compiler, available from Synopsys Inc. of Mountain View, California.

In step 102 Logic synthesis algorithms optimize the network by changing the structure of the network without changing the function of the network.

In step 103, logic network synthesis algorithms map the abstract network representation to cells from the library.

Step 104 optimizes the size of the cells. The size of the cell, together with the load of the cell determines the delay of the cell. The sizing algorithm adjusts the sizes of the cells. Changing the sizes of the cells affects delay and area, and the sizing algorithm manipulates the cell sizes so as to minimize delay and area.

Steps 102, 103 and 104 are performed by a computer program, such as "Design Compiler" TM available from Synopsys Inc., of Mountain View, California.

Step 105 determines the placement of the cells on the chip. Placement algorithms attempt to keep the length of the nets short, as longer nets need more area on the chip and the increased net load of longer nets will make the network slower. The network remains unchanged during the placement.

Finally step 106 determines the exact routing of the nets on the chip. Steps 105 and 106 are done by a computer program, such as "Cell Ensemble" TM available from Cadence Inc. of San Jose, California.

(Problems with prior art)

The major problem with the conventional approach is that the net length and hence the cell delay is not known until after placement. Before placement, net length must be estimated. This is usually done with an estimation function or table which gives the load of a net based on its fanout. Experience has shown that it is very difficult to estimate the length of the nets accurately. Essentially net length behaves as a random variable.

The result is unpleasant surprises after placement step 105. Some nets turn out to be longer than expected, and because of the longer delays the timing constraints are not met. Timing closure is not certain until after step 105.

If timing closure is not achieved the options the designer has are expensive and unreliable. He may choose to fix the design manually, which is difficult and time consuming, because the automatically optimized network is hard to understand. He may choose to change his HDL specification and repeat the synthesis process. Again timing closure will not be certain until after placement, which means that the entire process needs to be traversed before the designer knows if his HDL changes were successful.

A common method of dealing with inaccurate net load estimates is to use net load estimates which are considerably larger than accurate estimates. This causes the sizes of the cells to be considerably larger than necessary but reduces the probability of not meeting the timing constraints after placement. Clearly using cells with sizes which are larger than necessary is wasteful in both silicon area and power consumption. The chips thus synthesized will be larger, cost more to produce and use more electrical power than necessary.

A second problem with the conventional approach is that the effect

of synthesis decisions is hard to calculate. Performing timing analysis during optimization is very time consuming, and accounts for most of the run time of conventional synthesis systems.

In step 103 it is difficult to take decisions based on delay without knowing load and size of the cells as well.

In step 104, changing the size of a cell affects the loads of the fanin cells, and thus the delay of the fanin cells. In more complex delay models, which take into account the transition time of the signals, also the delay of the fanout cells is affected. Usually the size parameter cannot have any arbitrary value. Because the library of cells has been designed before the network synthesis started, only a few sizes are available. 3 or 4 sizes per cell is common. This makes it harder to find a good solution.

In step 105 the placement program will modify the net lengths. Depending on which location was chosen for each cell, the length of each net can be different. As the length differs, the capacitive load of the net changes, and as a result, the delay of the cell driving the net changes. Therefore the delays which were carefully optimized during the logic synthesis, are very different after placement, and the optimization of the network is not very good.

Much of the progress in the state of the art can be characterized as increased integration. This is represented in figure 1 as various feedback paths, which repeat and alternate steps. The general direction has been towards programs which do structuring, mapping, sizing and placement simultaneously. It has lead to increasingly complex software systems which are slow and difficult to design and maintain.

Iterating between placement and sizing has been especially hard to execute because placement programs are not sold by the same design automation software vendors as logic synthesis programs. Also they are not run by the same users: the logic synthesis program is often run by the designer, who also wrote the HDL specification. The placement program is often run by the silicon chip manufacturer, after the design is considered complete.

SUMMARY OF THE INVENTION

(objectives)

It is an object of the present invention to achieve timing closure as quickly as possible in the synthesis process. This will give the human designer early warning if the design is infeasible, because it is over constrained.

The invention achieves this object by

- speeding up the algorithms in the synthesis process
- performing the feasibility check early in the synthesis process, that is, before placement, instead of after placement.
- maintaining feasibility throughout the remainder of the synthesis process, so that it can be guaranteed to succeed and can be executed automatically.

The present invention overcomes the problems of the conventional approach by not choosing a size for a cell at all. Rather than choosing a default size, as conventional methods do, we choose a delay and let the size implicitly be whatever it needs to be to meet that delay.

In the conventional method of optimization, the structure, mapping, size and placement are chosen to optimize delay and area. In our formulation of the problem, we choose the structure, mapping, delay and placement, to optimize size and area. In our formulation, size only affects the area, so area only remains as an optimization goal.

The present invention speeds up the programs by simplifying or eliminating timing analysis. While optimizing network delay in order to achieve timing closure, the delays are constant, which will speed up delay calculation. Also, since changes to the network do not change the delay of the cells, the amount of recalculation is drastically reduced. While optimizing area after timing closure has been achieved, timing analysis is not needed, as the delay of the cells does not change. The one step where timing analysis using complex delay models is necessary is in the stretching step. Here too, library design rules are taken into account.

The present invention maintains timing closure after it has been achieved by adjusting the size of the cell during or after placement. The adjustments compensate for the fact that the placement algorithm can assign different net lengths to different nets and that these lengths are difficult to predict before placement.

(language of the main claims)

In accordance with the purpose of this invention, as embodied and broadly described herein, the invention is a method for the structuring and mapping of an unmapped digital network comprising the following steps:

- a) structuring of the digital network, using network slack as an optimization goal, where network slack is calculated assuming that the delay of the cells of the network is constant with respect to load.
- b) mapping of the cells of the network, using network slack as an optimization goal, where network slack is calculated assuming that the delay of the cells of the network is constant with respect to load.
- c) estimation of the area of the network based on net load

In further accordance with the purpose of this invention, as embodied and broadly described herein, the invention is a method for the placement and sizing of cells of a mapped digital network, the method comprising the steps of:

- d) Choosing a target delay for each cell.
- e) Computing the network slack using the target delays.
- f) Placement of the cells of the network.
- g) Sizing of the cells of the network such that the network meets the network slack as computed by step b).

In further accordance with the purpose of this invention, as embodied and broadly described herein, the invention is a method for the placement of the cells of a digital network, the method comprising the steps, of:

- f1) The calculation of net weights that reflect the change of network area due to sizing as a function of net length.
- f2) Placement of the cells of the network where the weighted network net length is used as a placement objective, the weighted network net length being the sum of the weighted net lengths of all nets, each net length being

multiplied by a weight,

Objects and advantages of the invention will be set forth in part in the description which follows and in part will be obvious from the description or may be learned by practice of the invention. The objects and advantages of the invention will be realized and attained by means of the elements and combinations particularly pointed out in the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute part of this specification, illustrate several embodiments of the invention and, together with the description, serve to explain the principles of the invention.

Fig. 1 is a flow chart showing the flow of the conventional method.

Fig. 2 is a flow chart showing the flow according to the present invention.

Fig. 3 is a block diagram of a computer.

Fig. 4 is a schematic diagram of a digital network.

Fig. 5 is a schematic diagram of the electronic and the physical implementations of a cell.

Fig. 6 is a timing diagram illustrating the concept of slack and other timing concepts.

Fig. 7 is a graph showing the relationship between the delay of a cell, the size of a cell and the load of a cell.

Fig. 10 is a

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Reference will now be made in detail to the preferred embodiments of the invention, examples of which are illustrated in the accompanying drawings. Wherever possible, the same reference numbers will be used throughout the drawings to refer to the same or like parts.

(what insight led to this invention?)

(Software patent application)

The preferred embodiment of the present invention is as one or more computer programs. The digital network, its elements and attributes, exist during the process of the method only as data structures in the memory of the computer. Methods in prior art are known to persons of ordinary skill in the art to convert the design data in memory eventually to an actual physical implementation of the network.

(Computer system)

Figure 1 is a block diagram of a computer system 100 in accordance with

the present invention. Computer system 100 includes a central processing unit 101, bus 102, memory 103, input device 104 and output device 105. It will be understood by a person of ordinary skill in the art that computer system 100 can also include numerous elements not shown in the figure for the sake of clarity, such as disk drives, tape drives, mice, printers, network connections, additional CPUs, etc. Memory 103 contains a program 107, which embodies the invention, and a data structure representation of the network 106.

(Network terminology)

Figure 2 is a schematic diagram of a digital network. Digital *network* 200 is composed of a plurality of *cells* 205, 206, 207, 208, 209, connected by a plurality of *nets*. Each cell (e.g. 208) has one or more inputs 212, 213, and a single output 214. Each net is connected to one output and one or more inputs. The cells can be combinational *gates* 207, 208, 209, whose function is represented as an expression in the Boolean algebra, using AND, OR and NOT operators, or the cells can be registers 205, 206. All feed back loops 210, 211 in the network contain at least one register. Cells whose outputs are connected to the inputs of a cell are called the fanin of the latter cell. Cells whose inputs are connected to the output of a cell are called the fanout of the latter cell.

(Computation)

The digital network performs a logic *function* by processing digital binary input data in a number of cycles. The input data is presented to the network on its *primary inputs* 201, 202, and the result of the computation of the network function is presented at the *primary outputs* 203, 204, of the network. The computation of the function takes one or more cycles. During each cycle the gate functions are calculated. The results are stored in the registers for use in the next cycle. The data flow is from inputs to outputs and will be assumed to be pictured as going from left to right throughout this text.

(Mapping)

Each cell can be *mapped* to a *book* in which case a electronic realization in terms of transistors has been chosen, and physical attributes such as delay are known. If a cell is not mapped to a book it is *unmapped*, in which case no electronic realization has been selected, and the function of the cell is only known in abstract terms, such as Boolean algebraic expressions. A *library* of books is designed in advance, before synthesis starts. These books are of generic types, and can be used to build arbitrary designs.

(Size)

The cell has a delay, an area and its input pins have an input pin capacitance. The *size* of the cell is a multiplier which is applied to equally to all transistor channel widths in the electronic circuit of the cell. Thus size of a cell is a scale factor which is used to scale its output load driving capability (see below), its area, and its input pin loads.

(Timing constraints)

Each primary input or primary output has an associated delay,

called the **input delay**, respectively **output delay**, which represents delays external to the network. The network needs a certain amount of time to perform one cycle, called the **cycle delay**. Together the cycle delay, the input delays and the output delays form the timing constraints of the network. Meeting the timing constraints is called **timing closure** and it is a major objective of the synthesis process.

(Timing)

The delay of a path is measured as the sum of the gate delays over said path from begin point to end point. The cycle delay is the maximum of all path delays. Primary outputs and register inputs are timing path end points. Primary inputs and register outputs are timing path begin points.

(Slack)

The **arrival time** of the data at a gate is computed by taking the maximum arrival time of its fanin cells each increased by the delay from the input pin to the output pin. The arrival times can be computed by traversing the network from left to right, that is, from timing begin points to timing end points. Similarly the **required time** of the data at a gate can be computed by traversing the network from end points to begin points. The required time is the minimum required time of its fanout cells, each reduced by the delay from the input to the output pin of that fanout cell.

The difference between the required time and the arrival time is the **slack**. If the arrival time is smaller then the required time, the timing constraints are met, and the slack is positive. If the arrival time is larger then the required time, the timing constraints are not met, and the slack is negative. The arrival time and required time may be different depending on whether the data is zero (0) or one (1). There also may be multiple arrival times and multiple required times to model a variety of timing constraints. All slacks can be summarized as a single worst slack number, called the **network slack**. Timing closure is achieved if the network slack is non-negative.

(Delay model)

The delay D of a gate depends on many factors, among them its function, its size S and the capacitive load C of the gate. The delay may be different for different inputs of the gate and it may be different for the falling and the rising transition. It is important to note that the dependency on size and load can be captured as the dependency on a single parameter C/S , and the delay D is non-negative and monotonically increasing with C/S .

$$\begin{aligned} C &> 0 \\ S &> 0 \\ D &= f(C/S) \\ f(C/S) &> 0 \\ f'(C/S) &> 0 \end{aligned}$$

Figure 4 illustrates the relationship between the three variables, size S , delay D and capacitive load C . Each of the three planes shows the relationship between two variables, while the third variable is constant, (not necessarily zero).

(Detailed description)

The preferred embodiment of the invention is a software program that can be stored in the memory of a computer, and can be executed by the central processing unit of the computer so that the computer performs the method described herein.

The software program consists of many parts or subprograms which together perform the method described in this invention. The essence of the invention is that logic synthesis is done in a size independent way, and that sizes are determined after placement, and that we guarantee that that the delay numbers before placement can be met by sizing after placement.

The preferred embodiment consists of three parts: (See fig...) to wit

- a) The logic synthesis program
- b) The placement program
- c) The sizing program

- analyze the library
- read logic
- library independent optimization.

The first step is to analyze the library that will be used for logic synthesis. The library contains the cells that will be used to implement the logic function. Contrary to the standard method of performing logic synthesis, we will assume that each cell can be sized by a continuous, positive real variable S , which increases both the load driving capability of the cell and the area linearly. In other words, the area of a cell is $S \cdot A$ and the delay of a cell is $D = f(C/S)$.

The library analysis will determine a good value for C/S for each cell in the library. Using this value, it determines a constant delay for each gate.

(choice of C/S)

Since the library analysis is not dependent on the actual network being synthesized, library analysis can be performed before beginning the synthesis process. We will now continue to describe the actual automatic synthesis process, beginning at the with reading the design. The design is expressed in a high level design specification language, for example VHDL or Verilog, and is syntactically parsed and transformed into a logic network representation by a computer program, such as HDL Compiler, available from Synopsys Inc. of Mountain View, California.

(library independent optimization)

Initially the network is library independent and library independent optimizations are performed. Mostly these optimizations change the structure of the network, and the Boolean functions of the cells, without changing the overall function of the network. The types of optimizations that should be performed are behavioral optimizations, such as resource sharing, sequential optimizations such as retiming, algebraic optimizations such as kernel extraction and Boolean optimizations such as redundancy removal. There is a large amount of literature on how each of these classes of optimizations can be performed.

(mapping for delay)

Following the library independent optimizations the network is mapped to a library of cells. This means that the logic functions of the cells are implemented with actual cells from the library. During this process the A large body of literature exists already on the subject of mapping digital networks. The preferred embodiment would use a previously published algorithm such as

(post mapping optimizations)

Due to restrictions in run time, it is impractical to explore the entire design space during the mapping algorithm. Necessarily, the mapping algorithm has to ignore many possible solutions because either they are unlikely candidates or they are very similar (but not identical) to other, considered solutions. In addition, in the constant delay approach, it is easy to evaluate the impact on timing of synthesis decisions, but it is much harder to evaluate the impact on the total network area. Therefore the mapping algorithm necessarily cannot accurately optimize area.

(pin swapping)

An example of candidate mappings which are not explored during mapping because they are too similar to other mappings follows here: Often gates have several pins which are functionally interchangeable. For instance for a 4 input NAND gate, there are $4! = 24$ possible permutations for the 4 input pins. Usually these pins are not equally fast, because of an inherent asymmetry of the electronic circuit. Because the differences are small, it is not worthwhile to consider all of these different of different permutations during mapping. It is more efficient to pick one arbitrarily and to select the best permutation of the inputs after mapping.

(structuring - boundary move)

Using constant delay it is considerably easier to predict the effect of a change to the network than with the conventional delay models. This can be used to do timing optimization by means of restructuring after technology mapping has been done. For example, we can use the "boundary move" transformation to reduce the number of levels in the logic in the mapped network. The boundary move transform, illustrated in fig x, reduces the number of levels by bringing connection x forward. To make the change legal it is necessary that gates x, y and z are fanout free. If not, they must be made fanout free by making a copy. In the conventional approach to logic synthesis, copying logic will increase the load on gates x, x, x and therefore increase the delay. To predict if the transformation will improve delay, or hurt delay, it was necessary to run a complete static timing analysis with accurate delay models. If the change actually worsened the delay, then the change would be undone.

In the constant delay model approach, the effect of this change can be easily predicted. Note that changes in loads do not affect delay. The only change that affects delay, is the change of the fanin of gate xxx. The delay can easily be predicted by simple addition of gate delays.

(area estimation)

To perform area optimization it is necessary to calculate the sizes of the cells. The sizes can be calculated in a straightforward manner from the loads. The loads are calculated by adding the net load and the pin load. The net load consists of the load of the net, which can be estimated using a conventional net load model, plus any other fixed load, such as the load of a primary output. The pin load is not fixed, that is, the load of an input pin depends on the size of the cell. This creates a dependency: To calculate the load of a cell, we need to calculate the size of its fanout cells. Therefore the algorithm starts calculating as far downstream as possible, and traverse the network in a direction opposite to the flow of data. In a combinational network this can be achieved by starting at the primary outputs and traversing the network in a levelized order towards the primary inputs. In a sequential network there may be one or more loops, resulting a cyclic dependency: there is no rightmost cell. In this case the computation can start anywhere in the cycle, and repeats the cycle several times, until the capacitances converge and the error is sufficiently small.

It is possible that this iteration will not converge and that the capacitance will increase in every iteration, by progressively larger amounts. This situation is detected by requiring the increment to be smaller than a preset maximum after a fixed number of iterations. The iteration does not converge if the network is an infeasible solution: The current network cannot be expected to work at this speed because its gain is too small. Changes need to be made to the network to increase the gain, which will usually mean increasing the delay of the network as well.

After the loads have been calculated the size can be calculated by dividing the actual load by the predetermined typical load. The input capacitance can be calculated by multiplying the unit gate input capacitance by the size. The ratio of these numbers is the size of the gate. The size is a scale factor, which can be applied to the area of the gate, to give the area of the sized gate. The area of the network can be estimated as the sum total of the areas of the sized gates, plus the net area as estimated from the total length of all nets.

(net weights)

Various algorithms, such as buffering and placement optimize the network by manipulating the loads in the network. Placement optimizes the net length, which directly related to the net load, and buffering reduces the load on a cell by adding extra delay. These algorithms can benefit from a more efficient calculation of the effect that changing the load of a cell has on sizing. We can do this by calculating single parameter per net, called the net weight, which represents the sensitivity of the total area of the network with respect to the load on that net.

This net weight can be calculated in a manner that is very similar to the calculation of the pin loads during the area calculation above. Starting at the primary inputs, the net weight of the first (left-most) gate is equal to its area per unit load. The net weights of the other cells can now be calculated with a recurrence relation traversing the network from left to right.....

to the calculation of the loads. Starting at the primary inputs the network to 0. Set the net load on the net in question to 1. Perform the iteration and calculate the area as described in the previous section.

Since all calculations are linear, the effects simultaneous changes in loads of several nets can be superimposed, that is, added together.

(buffering)

The next step in the synthesis process is the buffering step. The buffering algorithm adds buffers to the network guided by the timing analysis and the area analysis. In the constant delay model, the buffers have a fixed delay, and thus the impact of inserting a buffer is easily determined by subtracting the delay of the buffer from the slack. Thus the effect of adding a buffer on delay is always negative: a buffer always adds delay, never reduces delay. The main effect of adding a buffer is to save area because the source gate can be smaller because the it's load is smaller. The effect on area and gain can be determined by area analysis. Net weights.....

The buffering algorithm works as follows: First it finds locations in the network where a buffer can be added without increasing the network delay. This is done by subtracting the delay of the buffer from the local slack, to give the predicted slack after buffer insertion. If the predicted slack is larger than the network slack, then a buffer can be inserted without increasing the network delay. Next we have to calculate the reduction in load of this net, and check that area that is added by adding the buffer does not exceed the area saved by sizing down the source gate. The area added by inserting the buffer is simply the area of the buffer times its size, where the size is determined by the load on the buffer divided by the typical load of the buffer. The area saved by inserting the buffer can be calculated by first calculating the change in load due to the insertion of the buffer: some sinks are removed, the input load of the buffer is added, and the net load estimate may change as a result of the number of fanouts of the net changing. Using the net weight, we can estimate the impact on the network area. If the impact is positive (reduced area) the buffer is inserted. After the buffer is inserted, the capacitances need to be updated in the fanin cone of the buffer, while the net weights need to be updated in the fanout cone of the buffer.

(stretching)

The next step is the process of "Stretching" and "Compressing" the delays of the individual gates to meet the timing constraints. Gates which are on long paths which do not meet the delay constraint are "compressed" until the path does meet the timing constraint. Gates on the short paths which easily meet the timing constraints are "stretched". Gates with stretched delays require less area for the same load. In this step the delay of the gates is traded against the gain of the gates. When the delay of a gate decreases so does the gain of the gate. It is important that there is enough gain in the network.

Note that this can be done entirely independent of the sizes of the gates. Sizes of gates are not determined until much later in the process.

The stretching algorithm has two phases. In the first phase it will compress the delays of cells on long paths to meet timing constraints. In the second phase it will stretch the delays of the

gates on short paths to save area. For the purpose of stretching and compressing registers can usually be considered to be part of a path which they originate, but not of a path that they terminate. The stretching algorithm considers the cells on a path by path basis, processing the path with the smallest slack first. The delay of each cell on the path is adjusted by an amount which is equal to the slack divided by the number of stages on the path. After a cell has been adjusted, it becomes "locked" and its delay cannot be changed by the stretching algorithm. Stages which are locked are not counted when calculating the adjustments. For the stretching phase, the algorithm continues as above. In this phase the delays of the cells are increased, not decreased. The path that we work on is not the worst path, but it is the worst path with a slack greater than 0. (All other paths now have a slack of 0).

(Rule based stretching)

(incremental rule based)

(placement)

In our process a conventional placement method is augmented to optimize the area of the placed network. All placement methods known work by gradual refinement of the placement. Periodically, during the placement process, we recalculate the estimated net lengths, using the most recent, accurate placement information. From the net lengths, it calculates the sizes of the cells in the network. (See area estimation). The updated sizes can then be used for further placement and for more accurate net length calculations.

Placement primarily manipulates the lengths of the nets. Using the net weights, the area of the network can efficiently be estimated.

(final or discrete sizing)

Other embodiments will be apparent to those skilled in the art from consideration of the specification and practice of the invention disclosed herein. It is intended that the specification and examples be considered as exemplary only, with a true scope of the invention being indicated by the following claims.

WHAT IS CLAIMED IS:

(CONSTANT DELAY SYNTHESIS)

1. A method for the structuring and mapping of an unmapped digital network comprising the following steps:

- a) structuring of the digital network, using network slack as an optimization goal, where network slack is calculated assuming that the delay of the cells of the network is constant with respect to load.
- b) mapping of the cells of the network, using network slack as an optimization goal, where network slack is calculated assuming that the delay of the cells of the network is constant with respect to load.
- c) estimation of the area of the network based on net load

(typical load)

3. The method of claim 1, the delay of a book is used for the delay of each cell, the delay of the book being chosen by choosing a load size ratio C/S for each book, which is independent of the network.

4. The method of claim 3, where the delay of a book is also determined by the choice of the input transition time, which is independent of the network.

(continuous buffering assumption)

5. The method of claim 3,

where a parameter C/S for each book

is chosen to have the largest

possible value such that a long chain of cells of identical books each cell in the chain having identical value of parameter C/S, said chain cannot have simultaneously improved delay and improved gain by adding a buffer at some point to the same chain, even when the parameter C/S is chosen optimally after adding the buffer.

(buffering)

6. The method of claim 1, with the additional step of buffer insertion before step c), the buffers being inserted on paths with positive slack, as determined by subtracting the delay of the buffer from the slack of the path.

(more buffering)

7. The method of claim 6, where the buffer is inserted if area is saved.

8. The method of claim 7, where the area savings are estimated using net weights which reflect the change of network area due to sizing as a function of net length.

9. The method of claim 8, where the calculation of the net weights

is performed by starting at the primary inputs and traversing the network in the direction of the data flow, while calculating the net weight of a cell by summing over the fanin of the cell the product of the net weight of the fanin cell divided by the gain plus the cells area/load sensitivity.

as $W_i = \sum_j W_j / g_{ij}$

(stretching)

10. The method of claim 1, where step a) is comprising of:

a1) choosing a delay for each book

a2) the delay of the cell assuming the delay of its book

a2) adjusting the delay of each cell based on the slack

11. The method of claim 10, where in step a2) the delay of each cell is adjusted equally among the stages which have the same slack.

12. The method of claim 10, where in step a2) the delay is adjusted on each path, such that the slack of each path becomes 0.

(globally optimal mapping)

13. The method of claim 1, where in step b) the mapping is performed in two steps:

b1) a traversal of the network from

primary inputs and registers from left to right, while choosing at each cell the fastest matching books from all available matching books, using the constant delays of the books and the fastest arrival times of the fanins

of the matching book.

b2) a traversal of the network from right to left, while choosing at each cell the fastest matching book from the candidates selected during the previous traversal.

(Area estimation)

14. The method of claim 1, where in step c) consists of the following steps:

- c1) estimation of the net length based on the number of fanout cells.
- c2) estimation of the capacitive load of the cells using the net length
- c3) calculation of the sizes from the capacitive load.
- c4) calculation of the network area by summation of the product of the book area times the cell size.

(sizing algorithm)

15. The method of claim 14, where step c2) is performed by starting at the primary outputs and traversing the network in the direction opposite to the data flow, while calculating the load of a cell, by summing over the fanout of the cell, the product of the load of the fanout cell divided by the gain plus the net load of the cell.

(iterate till convergence)

16. The method of claim 15, where the network has loops, the traversal is started at primary outputs and arbitrary cells in the loop, and the calculation is repeated until convergence.

17. The method of claim 16, where the traversal is started at primary outputs and registers.

(Area optimization)

18. The method of claim 1, where steps a) and b) also use network area as an optimization goal, in addition to network delay, the network area being estimated as in step c).

(retiming)

19. The method of claim 1, where the structuring step is preceded by a retiming step, where registers are moved in the network, while preserving the function of the network, and where change to the network do not affect the delay of the individual cells.

(I could produce a bunch more of these sort of claims)

(TIMING CLOSURE)

20. A method for the placement and sizing of cells of a mapped digital network, the method comprising the steps of:

- d) Choosing a target delay for each cell.
- e) Computing the network slack using the target delays.
- f) Placement of the cells of the network.
- g) Sizing of the cells of the network such that the network meets the network slack as computed by step b).

(sizing for better placement)

21. The method of claim 20, where step f) is performed in gradual steps, each step being followed by a sizing step g)

(partitioning)

22. The method of claim 21, where step f) is performed by repeated partitioning steps, partitioning the cells in the network into two or more groups, each group being assigned to an subdivision of the plane, alternating the partitioning steps with sizing steps, essentially similar to step g)

(iterative improvement)

23. The method of claim 21, where step f) is performed by choosing an arbitrary initial location in the two dimensional plane for each cell, the placement being optimized by repeatedly changing the location of one or two cells at a time, while performing a sizing step, essentially similar to step g) after each location change.

(sizing)

24. The method of claim 5, where step g) is consists of the following steps:
g1) calculation of the net length based on the available placement information
g2) calculation of the capacitive load of the cells using the net length
g3) calculation of the sizes from the capacitive load.

(sizing algorithm)

25. The method of claim 24, where step g2) is performed by starting at the primary outputs and traversing the network in the direction opposite to the data flow, while calculating the load of a cell, by summing over the fanout of the cell, the product of the load of the fanout cell divided by the gain plus the net load of the cell.

(iterate till convergence)

26. The method of claim 10, where the network has loops, the traversal is started at primary outputs and arbitrary cells in the loop, and the calculation is repeated until convergence.

27. The method of claim 26, where the traversal is started at primary outputs and registers.

(cell generation)

28. The method of claim 20, where following step g) the layout of the cells is generated automatically to yield the exact transistor sizes calculated by step g)

(discrete sizing)

29. The method of claim 24, where step g3) consists of selecting the most suitable size from a limited set of available sizes.

(Weighted Placement)

30. A method for the placement of the cells of a digital network, the method comprising the steps, of:

f1) The calculation of net weights that reflect the change of network area due to sizing

as a function of net length.

f2) Placement of the cells of the network

where the weighted network net length is used as

a placement objective, the weighted network net length being the sum

of the weighted net lengths of all nets, each net length being multiplied by a weight,

(Calculation of net weights)

31. The method of claim 30, where step f1)

is performed by starting at the primary inputs and traversing the network in the direction of the data flow, while calculating the net weight of a cell by summing over the fanin of the cell the product of the net weight of the fanin cell divided by the gain plus the cells area/load sensitivity.
as $W_i = \sum_j W_j / g_{ij}$

(iterate till convergence)

32. The method of claim 31, where the network has loops, the traversal is started at primary inputs and arbitrary cells in the loop, and the calculation is repeated until convergence.

33. The method of claim 32, where the traversal is started at primary inputs and registers.

(Weighted Placement for power)

34. A method for the placement of the cells of a digital network, the method comprising the steps, of:

f1) The calculation of net weights that reflect the change of network power due to sizing as a function of net length.

f2) Placement of the cells of the network where the weighted network net length is used as a placement objective, the weighted network net length being the sum of the weighted net lengths of all nets, each net length being multiplied by a weight,

(Calculation of net weights)

35. The method of claim 30, where step f1) is performed by starting at the primary inputs and traversing the network in the direction of the data flow, while calculating the net weight of a cell by summing over the fanin of the cell the product of the net weight of the fanin cell divided by the gain plus the cells power/load sensitivity.
as $W_i = \sum_j W_j / g_{ij}$

(iterate till convergence)

36. The method of claim 31, where the network has loops, the traversal is started at primary inputs and arbitrary cells in the loop, and the calculation is repeated until convergence.

37. The method of claim 32, where the traversal is started at primary inputs and registers.

38. The method of claim 35, where the power/load sensitivity of a cell is calculated as the product of the switching frequency at that cell times the square of the voltage times the capacitance, where capacitance is calculated as the sum of the net load plus the internal capacitance of the cell, scaled with its size.

ABSTRACT

A method for the design of digital networks consisting of a plurality of cells. The invention uses constant delays during logic synthesis and sizes the cells after placement so as to meet the cycle delay predicted before synthesis. The method chooses a constant delay before logic synthesis and guarantees that it can maintain this delay after placement by means of sizing. Thus it overcomes the unpredictable effects of placement on the cycle delay. The invention also describes methods for choosing the sizes of the gates and a method for inserting buffers.

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.